

# UNCLASSIFIED

AD NUMBER
AD919431
NEW LIMITATION CHANGE
TO Approved for public release, distribution unlimited
FROM Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; 5 Jun 1974. Other requests shall be referred to Commander, Frankford Arsenal, Attn: SARFA-FCW-D, Philadelphia, PA.
AUTHORITY
FA D/A ltr, 2 Oct 1974

THIS PAGE IS UNCLASSIFIED

AD919431

FINAL REPORT  
SUPPLEMENT TO PARAMETRIC STUDY OF  
ADVANCED FORWARD AREA AIR  
DEFENSE WEAPON SYSTEM  
(AFAADS)

VOLUME II DATA PROCESSING  
REQUIREMENTS ANALYSIS

31 January 1974

Submitted to:

U. S. Army, Frankford Arsenal  
Bridge and Tacony Streets  
Philadelphia, Pennsylvania 19137

Prepared Under Contract: DAAA25-73-C0373  
(formerly DAAG05-70-C0328) MOD P 00011

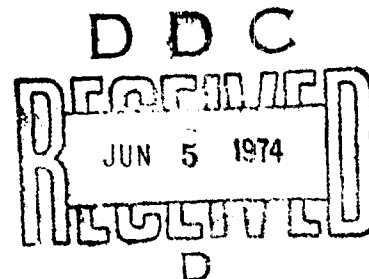
Distribution limited to U.S. Gov't. agencies only;  
Test and Evaluation; 5 June 1974. Other requests  
for this document must be referred to Commander,  
Frankford Arsenal, ATTN: SARFA-FCW-D,

Prepared by:

Data Systems Division  
Litton Systems, Inc.  
8000 Woodley Avenue  
Van Nuys, California 91409

Best Available Copy

TD 40001



## ACKNOWLEDGEMENTS

Litton is happy to acknowledge the excellent guidance and assistance provided on this contract by Mr. Robert Pfeilsticker, Mr. Walter Ryba, Mr. Bernard Kirshtein, Mr. Stanley Goodman, and Mr. Kenneth Heulitt of Frankford Arsenal.

Best Available Copy

## **REPRODUCTION QUALITY NOTICE**

**This document is the best quality available. The copy furnished to DTIC contained pages that may have the following quality problems:**

- **Pages smaller or larger than normal.**
- **Pages with background color or light colored printing.**
- **Pages with small type or poor printing; and or**
- **Pages with continuous tone material or color photographs.**

**Due to various output media available these conditions may or may not cause poor legibility in the microfiche or hardcopy output you receive.**



**If this block is checked, the copy furnished to DTIC contained pages with color printing, that when reproduced in Black and White, may change detail of the original copy.**

## P R E F A C E

This Volume II of the Advanced Forward Area Air Defense System (AFAADS) Phase III final report presents an overall description of the conceptual design and sizing for a digital AFAADS gun fire control computer, including the operator display and control panel. It is submitted in fulfillment of Tasks 1D and 1E of the Statement of Work under Modification P00011 of Contract DAAA25-73-C0373 (formerly Contract DAAG05-70-C0328). (See Appendix A)

The material of this volume is divided into three parts:

Part I, Summary, is a brief overview of the conceptual design for the digital AFAADS gun fire control computer.

Part II, Overall System Description, presents an expanded description of the conceptual design. Estimates for the development and production of this computer are also given.

Recommendations for immediate follow-on activities to this study are included.

Part III, Analysis of the Fire Control System, provides the basis for the conceptual design. The algorithms used in the conceptual design are detailed. The necessary tactical and computational parameters are listed. The software concepts are developed, and the hardware conceptual design is presented.

Thus, the volume provides a complete and stand-alone description of the requirements and conceptual design for a digital AFAADS gun fire computer. The logic and algorithms of the fire control solution are based upon the analyses presented in Volume I of this report and in the final reports to Phases I and II of AFAADS (References 1 and 2).

Best Available Copy

## FOREWORD

This report describes the research effort of the Data Systems Division of Litton Systems, Inc., under Modification P00011 to Contract DAAA25-73-C0373, (formerly DAAG05-70-C0328). The objective of this work was to provide additional analytic and simulation effort in support of the parametric analysis of predicted fire air defense systems.

The report is presented in two volumes. Volume I, Analysis, by Herbert K. Weiss reports the analytical effort and the simulation verification procedures. Section 9 on simulation verification was made possible by simulation flowgraphs of the Ginsberg simulation developed by Mr. Barry Seid. Simulation runs were made by Mr. Jacky Onishi. Mr. F. V. Wilson provided analytical support for Section 7.0 on Countermeasures.

Volume II, Data Processing Requirements Analysis, is based on an analyses by Dr. Richard D. Young, Dr. Alfred J. Ess, Mr. Caesar F. Chavez and Mr. Herman A. Fischer.

Earlier effort under this contract at Litton is reported in a previously published series of five volumes,

## TABLE OF CONTENTS

<u>Section</u>		<u>Page</u>
	ACKNOWLEDGEMENTS	ii
	PREFACE	iii
	FOREWORD	iv
1	INTRODUCTION	1-1
	1.1 VOLUME OUTLINE	1-2
2	CONCLUSIONS AND RECOMMENDATIONS	2-1
	2.1 CONCLUSIONS	2-1
	2.2 RECOMMENDATIONS	2-2
3	OPERATIONAL DESCRIPTION	3-1
	3.1 TACTICAL EMPLOYMENT OF AFAADS	3-1
	3.2 THE AFAADS WEAPON SYSTEM	3-2
4	FIRE CONTROL COMPUTER CONCEPTS	4-1
	4.1 FIRE CONTROL SYSTEM LOGIC	4-1
	4.2 SOFTWARE DESIGN CONCEPTS	4-4
	4.3 HARDWARE DESIGN CONCEPTS	4-6
5	EXAMPLE OF AFAADS OPERATION	5-1
	5.1 OPERATING MODES	5-1
	5.2 TYPICAL FIRE MISSION	5-4
	5.3 OPERATOR CONTROL AND DISPLAYS	5-8
6	SYSTEMS ANALYSIS AND OPERATIONAL REQUIREMENTS	6-2
	6.1 BASIC FIRE CONTROL SITUATIONS	6-2
	6.2 COMPUTER LOGIC STRUCTURE	6-5
	6.3 BASIC PARAMETERS FOR THE AFAADS GUN SYSTEM	6-15

## TABLE OF CONTENTS (Continued)

<u>Section</u>	<u>Page</u>
6.3.1 Gun and Gun Mount	6-15
6.3.2 Angle Tracking Sensor	6-18
6.3.3 Range Tracking Sensor	6-21
6.3.4 Sensor Mount Servos	6-23
6.3.5 Vehicle Orientation Sensors	6-24
6.3.6 Digital Computer	6-25
<b>7 OPERATOR CONTROLS AND SCENARIOS</b>	<b>7-1</b>
7.1 THE OPERATOR'S DISPLAY PANEL	7-1
7.2 INITIALIZATION SCENARIO	7-4
7.2.1 Beginning Initialization	7-4
7.2.2 Initialization I and II	7-5
7.3 FIRE MISSION SCENARIO	7-9
7.3.1 Acquisition and Lock-On	7-10
7.3.2 Firing on Target	7-12
7.3.3 Cease-Firing and Disengagement	7-17
7.3.4 New Target Sighting	7-17
7.3.5 Passing Targets	7-17
7.4 OTHER MISSION SCENARIOS	7-18
7.4.1 Manual Mode	7-18
7.4.2 Test Mode	7-19
<b>8 DATA PROCESSING PERIPHERALS</b>	<b>8-1</b>
8.1 PERIPHERAL DEVICES	8-1
8.2 SPECIFIC INTERFACE SIGNALS	8-5

## TABLE OF CONTENTS (Continued)

<u>Section</u>		<u>Page</u>
9	SOFTWARE	9-1
9.1	INTRODUCTION	9-1
9.2	SOFTWARE PARTITIONING	9-1
9.2.1	Control Software	9-1
9.2.2	Applications Software	9-3
9.2.3	Software Test Program	9-12
9.3	ANALYSIS OF APPLICATIONS REQUIREMENTS	9-14
9.3.1	Processing Requirements	9-14
9.3.2	Memory Requirements	9-20
9.3.3	Conclusion Summary	9-24
9.4	PROGRAM PRODUCTION	9-24
9.4.1	Program Material	9-26
9.4.2	Language of Implementation	9-27
9.4.3	Documentation and Review	9-27
9.4.4	Programming Support Center	9-28
9.4.5	Field Employment Support	9-30
10	HARDWARE - AFAADS COMPUTER	10-1
10.1	KEY FEATURES OF THE AFAADS COMPUTER	10-1
10.2	AFAADS SYSTEM BLOCK DIAGRAM AND INTERFACES	10-2
10.3	COMPUTER REQUIREMENTS AND DESIGN OBJECTIVES	10-2
10.3.1	Cost Objective	10-2
10.3.2	Physical Requirements and Objectives	10-4
10.3.3	Input/Output Requirements	10-4

## TABLE OF CONTENTS (Continued)

<u>Section</u>	<u>Page</u>
10.3.4 Design Requirements for the Operator's Panel	10-5
10.3.5 Processing Objectives and Requirements	10-6
10.3.6 Maintainability and Logistic Support Concepts	10-7
10.3.7 Requirements for Training and Manuals	10-7
10.3.8 Environmental Requirements	10-7
10.4 COMPUTER BLOCK DIAGRAM AND PHYSICAL DESCRIPTION	10-8
10.4.1 Housing with Cover	10-11
10.4.2 Operator's Panel	10-11
10.4.3 Power Conditioner	10-16
10.4.4 Input/Output Logic Card	10-16
10.4.5 PROM Logic Cards	10-16
10.4.6 RAM Logic Card	10-17
10.4.7 Micro Control Logic Card	10-17
10.4.8 Mini-Control Logic Card	10-18
10.4.9 Data Logic Cards	10-18
10.4.10 Test Connector	10-19
10.5 COMPUTER OPERATION	10-19
10.5.1 Operator's Panel	10-19
10.5.2 Program	10-22
10.5.3 Micro Control	10-23
10.5.4 Interrupts	10-23
10.5.5 Interval Timer	10-25
10.5.6 Input/Output	10-25

## TABLE OF CONTENTS (Continued)

<u>Section</u>	<u>Page</u>
10.5.7 Hardware Tests	10-26
10.6 TRADE-OFFS	10-29
10.6.1 Program Storage	10-30
10.6.2 Operand Storage	10-30
10.6.3 Minicomputer Control	10-31
10.6.4 Push Buttons	10-32
10.6.5 Multiposition Switch - Few Positions	10-32
10.6.6 Multipurpose Switch - Many Positions	10-33
10.6.7 System Test Results	10-33
10.6.8 LED Display Code	10-34
10.6.9 LED Illumination	10-34
10.6.10 Indicators	10-35
10.6.11 Alarm	10-35
10.6.12 Input/Output Interface	10-36
10.7 GROWTH CAPABILITY	10-36
10.8 NEXT PHASE	10-37
10.8.1 Mechanical	10-37
10.8.2 Electrical and Circuits	10-37
10.8.3 Architecture	10-37
10.8.4 Logic Design	10-38
10.8.5 Parts Count/Cost Estimate	10-38
10.8.6 Build Prototype	10-38
10.8.7 Test	10-38

## TABLE OF CONTENTS (Continued)

<u>Section</u>		<u>Page</u>
11	SEMI-AUTOMATIC FIRE CONTROL ALGORITHMS	11-1
11.1	ALGORITHM SELECTION	11-1
11.2	PROCESSING FUNCTIONS	11-2
11.3	THE REAL TIME FIRE CONTROL ALGORITHMS	11-3
12	DEGRADED MODES	12-1
12.1	INTRODUCTION	12-1
12.2	NON-CRITICAL FAILURE	12-1
12.2.1	Types of Non-Critical Failures	12-1
12.2.2	Planned Corrective Action	12-2
12.3	CRITICAL FAILURES	12-4
12.3.1	Types of Critical Failures	12-4
12.3.2	Alternatives	12-5
13	TEST MODE	13-1
13.1	TEST CONCEPTS INVESTIGATED	13-1
13.2	TYPICAL SELF-TEST SEQUENCE	13-2
13.2.1	Initialization Procedures	13-2
13.2.2	Computer Confidence Self-Test	13-2
13.2.3	System Dynamic Test	13-4
13.2.4	System Dynamic Test Results	13-11
13.3	GROWTH IN SELF-TEST CAPABILITIES	13-12
<u>Appendix</u>		
A	EXCERPTS FROM STATEMENT OF WORK	A-1
B	AFAADS FINITE LENGTH TRACKING FILTERS	B-1
C	BALLISTIC CORRECTION FOR DOWN WIND	C-1

TABLE OF CONTENTS (Continued)

<u>Appendix</u>		<u>Page</u>
D	VALIDITY CRITERIA FOR FLIR/LASER PROJECTILE MISS DISTANCE DATA	D-1
E	SAMPLE BIAS CORRECTION ALGORITHM	E-1
F	BALLISTIC CORRECTION FOR GRAVITY DROP	F-1
G	BALLISTIC CORRECTION COMPUTATION FOR CROSS WIND	G-1
H	TABLE OF ACRONYMS	H-1
I	REFERENCES	I-1

## LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
3-1	AFAADS Air Defense Battalion	3-3
3-2	AFAADS System Block Diagram	3-5
4-1	AFAADS Operating Modes	4-2
5-1	AFAADS Engagement Geometry	5-5
5-2	Typical AFAADS Fire Mission Time Event Plot	5-6
5-2a	Typical AFAADS Fire Mission Time Event Plot	5-7
5-3	Target Range from AFAADS Gun	5-9
5-4	Target Azimuth from AFAADS Gun	5-10
5-5	Target Elevation from AFAADS Gun	5-11
5-6	Operator's Panel	5-12
6-1	AFAADS Operating Mode Logic Structure	6-6
6-2	Operating Situations Flow Chart	6-8
7-1	Operator's Panel	7-2
8-1	Input/Output Diagram	8-6
9-1	Executive Functions	9-2
9-2	Applications Software Modules	9-4
9-3	Real Time Fire Control Operations	9-7
9-4	Test Functions	9-13
9-5	Minimal Programming Support Center	9-29
9-6	Non-Minimal Programming Support Center	9-31
10-1	AFAADS Computer-Oriented Interconnection Block Diagram	10-3
10-2	AFAADS Computer Block Diagram	10-9
10-3	AFAADS Computer Mockup	10-10
10-4	Operator's Panel	10-15

# LIST OF ILLUSTRATIONS (Continued)

<u>Figure</u>		<u>Page</u>
10-5	Instruction Formats	10-24
11-1	Simplified AFAADS Real Time Fire Control Block Diagram	11-4
11-2	AFAADS Real Time Fire Control Block Diagram	11-5
11-3	Operator's Panel Fire Control Interface Block Diagram	11-6
11-4	Portion of the Target Acquisition Routine	11-11
11-5	Target Prediction and Ballistic Solution Logic	11-26
11-6	Initial Detection Routine	11-38
11-7	Initial Data Routine	11-40
11-8	Tracking Routine	11-42
11-9	Bias Muzzle Velocity Correction Logic	11-80
11-10	Logic Flow for Miss Distance Data Determination	11-86
11-11	Angular Bias Correction Logic	11-109
11-12	Definition of Transverse Miss Angles	11-112
13-1	Top Level Test Logic Flow Diagram	13-3
13-2	Medium Level Test Logic Flow Diagram	13-5
13-3	System Test Situation	13-7
13-4	System Test Firing Area	13-8
13-5	Coordinates for System Test Situation	13-9
13-6	System Test Results	13-13
A-1	Tentative Set of Modules for Computer Sizing	A-2
A-2	Flow Diagram for System Correction Based on Measurements of Projectile Miss Distances	A-4
A-3	Flow Diagram for Correction Based on Measurements of Projectile Miss Distances Using Synthetic Trajectory Reference	A-5
A-4	Computer Requirements Definition	A-6

# LIST OF ILLUSTRATIONS (Continued)

<u>Figure</u>		<u>Page</u>
C-1	Effect of Down Wind on the Projectile's Flight	C-2
D-1	Projectile Miss Distance Measurement Geometry	D-3
F-1	Geometry for Gravity Drop Correction	F-2
F-2	Oerlikon Superelevation Angle Versus Range and Flight Time	F-4
F-3	Superelevation Angle Vs. Target Elevation Angle	F-6
F-4	$\phi(s)/\cos E(p)$ Vs. $\cos E(p)$ at Constant Range	F-7
F-5	Superelevation Relationships at Constant Flight Time and Constant Range	F-8
F-6	Superelevation Vs. Time of Flight	F-10
F-7	Alternate Form for Superelevation Correction	F-11
G-1	Drift Correction for 10 m/sec. Cross Wind	G-2
G-2	Drift Correction Factor Vs. Elevation Angle	G-3
G-3	Maximum Range Draft Correction Factor	G-4
G-4	Drift Correction Vs. Time of Flight	G-5

## LIST OF TABLES

<u>Table</u>		<u>Page</u>
3-I	Principle AFAAD System Parameters	3-6
8-I	AFAADS Computer Peripherals	8-2
9-I	AFAADS Real Time Processing Load	9-17
9-II	Program Minicode Size	9-22
9-III	Data (Parameter) Memory	9-23
9-IV	Total Memory Requirements	9-25
10-I	Computer Input Signals	10-13
10-II	Computer Output Signals	10-13
11-I	Correlation Between Fire Control Block Diagrams	11-7
E-I	Target Flight Parameters	E-2
E-II	AFAADS Gun Fire Parameters	E-2
E-III	Bias Correction Constants	E-3
E-IV	Defined Matricies	E-4
E-V	Sample Bias Corrections	E-5

## PART I SUMMARY

The Data Systems Division of Litton Systems, Inc., submits this conceptual design and sizing analysis of a digital computer for an Advanced Forward Area Air Defense System (AFAADS). The analysis indicates that a small state-of-the-art modular stored program computer will perform all of the required tasks.

The specific air defense situation used in this sizing task is the defense of a forward area point target by a 35-mm rapid fire closed loop gun on a completely self-contained mount (tracked vehicle or towed). Variants on the chosen system will not significantly effect the computer design.

Each AFAADS mount would contain its own sensor system (FLIR/laser considered in this analysis but a millimeter radar could be substituted) mounted on a rate gyro stabilized head, twin barrel gun, magazine, servos, digital fire control computer, and prime power source. Control would be exercised by a single gunner/operator. The chosen 10,000 meter range sensor system would provide adequate warning for maximum range of fire of 4,500 meters against near sonic close support jet aircraft.

This AFAADS system, used in the concepts analysis, has a closed loop fire control capability in which sensed projectile misses are converted into gun pointing bias error corrections in azimuth, elevation, and muzzle velocity. Target prediction, as a part of the ballistic computations, includes energy conservation during a diving attack (transfer of potential energy to kinetic energy) and constant turn angle prediction. These, plus the more standard algorithms required for target tracking, ballistic computations, ballistic corrections, vehicle pitch and cant corrections, and sensor regeneration, are all used in determining the computer size.

The resultant AFAADS digital fire control computer weighs about 50 lbs. and has a volume of about 0.8 cubic feet including the operator's display and control panel. This 16-bit word machine has a programmable read-only memory (PROM) of 16,364 words, a random access core memory of 2,048 words, and a micro-control memory of 1,024 48-bit words. The real time semi-automatic fire control requires a 5 microsecond average instruction cycle time (slow due to the core memory) and approximately 7,600 instructions.

## PART I SUMMARY

(continued)

Capabilities are also included for a manual fire mode, operator self-testing in the field including an overall AFAADS system fire test, and operator inputs for ballistic corrections due to ammunition type, air temperature and density, wind speed and direction, and muzzle velocity; and a degraded mode capability.

Immediate follow-on work should include a refinement on the chosen processing algorithms, comprehensive flow charting of the software programs, and design detailing of the computer. This could then lead to development of a demonstration system.

## PART II OVERALL SYSTEM DESCRIPTION

### SECTION 1

#### INTRODUCTION

This volume, the Data Processing Requirements Analyses, of the Advanced Forward Area Air Defense System (AFAADS) Phase III final report, presents the software and hardware design concepts for a digital AFAADS gun fire control computer.

AFAADS is a generic analysis of a mobile (self-propelled or towed mount), self-contained, predicted fire, closed loop, anti-aircraft gun systems for use in low altitude forward area air defense against high speed close support aircraft, fixed wing and helicopter gun ships, and surface-to-surface battlefield cruise missiles. AFAADS will utilize a rapid fire gun of 20 mm to 40 mm caliber in semi-automatic fire.

For the digital computer concept studies it is assumed that:

- a) The vehicle is motionless when the gun is firing.
- b) Target acquisition system is provided.
- c) Automatic tracking and ballistic computation follow target detection and lock-on.
- d) FLIR/laser system makes up the tracking sensor.
- e) Actual firing of the gun remains an operator function.

Within the fire control system, AFAADS features closed loop predicted fire based upon projectile tracking. Emphasis is on point defense of forward area targets. Types of predicted fire include energy conservation diving attacks and constant angle turns, as well as simple linear prediction. Ballistic corrections include gravity drop; meteorological corrections for wind, air temperature, and air pressure; different types of projectiles; and muzzle velocity measurements.

Thus, a rather specific AFAADS gun system is postulated in order to develop a specific computer hardware and software conceptual design. It is believed that the parameters chosen are representative of any AFAADS type AA gun fire system. They are based upon the analyses of Volume I of this report and upon the results of the two earlier phases of this program.

(Ref 1 and 2). In choosing this specific system, care has been exercised in the conceptual design to meet other similar AFAADS gunfire system requirements.

#### 1.1 VOLUME OUTLINE

The volume is divided into 13 sections. The next four in Part II complete the overall system description as follows:

2. Conclusions and Recommendations, including follow-on task recommendations.
3. Operational Description, a brief summary of the envisioned tactical employment of AFAADS and of the overall weapon system.
4. Fire Control Computer Concepts, an overview of the fire control logic, algorithms, software, and hardware that make up the computer concepts.
5. Example of AFAADS Operation, a brief summary of the various operating modes, the displays and controls for the operator, and finally a time sequential description of a typical fire mission.

These sections are followed by Part II, Analysis of the Fire Control System. Here, in eight sections, the fire control requirements are analyzed in detail and the hardware and software concepts developed. The specific topics covered, by Section number, are:

6. System Analysis and Operational Requirements, covering the various operating modes, logic structure, and system parameters.
7. Operator Controls and Scenarios, detailing the functions available for control and monitoring and their operation in several scenarios.
8. Data Processing Peripherals, covering the interfaces with the FLIR/laser sensors, the gun mount, and other equipments.
9. Software, a discussion of the software concepts for implementing an AFAADS fire control solution.
10. Hardware-AFAADS Computer, the hardware concepts for the digital computer, design parameters, and physical and electronic characteristics.
11. Semi-automatic Fire Control Algorithms, a detailed presentation of the selected set of fire control algorithms for use in semi-automatic fire.

12. Degraded Modes, system capabilities under various levels and types of system degradation.

13. Test Modes, concepts for operational tests in the forward area.

Several supporting appendices follow.

## SECTION 2

### CONCLUSIONS AND RECOMMENDATIONS

The conceptual analysis and equipment sizing task shows that a modular, stored program, mini-computer can satisfactorily meet the operational requirements for AFAADS gunfire control against close support fighter aircraft and other forward area offensive air weapon systems. This computer utilizes current medium scale integration (MSI) state-of-the-art technology and can be readily expanded (or contracted) in size to meet additional (or lesser) fire control roles.

The conceptual design and computer requirements that have been developed are summarized in Subsection 2.1, Conclusions. This is followed in Subsection 2.2 with recommendations on the next phase in the development of an AFAADS digital fire control system.

#### 2.1 CONCLUSIONS

A digital fire control computer which meets the operational fire control requirements of an AFAADS gun system weighs about 50 lbs. and measures about 19 x 10 x 7 inches (0.77 cu. ft.). This computer is in a single unit with an integral operator's control panel as the front surface. A hinged cover provides protection during transit and can also act as a sun shield. Only two external connectors are required, one for prime power and one for the signal interfaces (input/output) with the tracking sensors, gun mount servos, and vehicle attitude sensors. Such a computer should be available for less than \$35,000; a very small fraction of the cost of a total AFAADS gunfire unit.

The real time fire control algorithms performed by this machine during semi-automatic gunfire cover the following functions:

- a. Target tracking to 10,000 meter range and 8,000 meter altitude.
- b. Target prediction to the impact point using one or more of the techniques for linear prediction, energy conservation in a diving attack, and acceleration prediction in a constant turn attack.
- c. Ballistic computation to 0.005 second accuracy out to the 4,500 meter maximum projectile range.

- d. Ballistic corrections for gravity drop, air temperature, air pressure, wind speed and direction, and muzzle velocity changes due to barrel wear. The ballistic effects of five different projectile types, including high velocity sub-caliber rounds, are also considered.
- e. Projectile miss distance measurement and processing with due consideration for target maneuvers during the projectile flight time.
- f. Gun pointing bias corrections in azimuth, elevation, and muzzle velocity based upon the measured miss distance.
- g. Sensor regeneration drive signals to keep the sensors on target if it should pass behind an obstacle or tracking is otherwise momentarily interrupted.

In addition, the computer can handle the various initialization computations, stand by during the manual fire mode, and perform various self-tests, both on itself and as a part of the overall gun weapon system.

The performance of these functions will be with a 16-bit word size machine of 5 microsecond average instruction cycle time. Real time operations are on a 100 millisecond maximum cycle time. Memory requirements are 16K for the program memory (programmable read only memory - PROM), 2K for the operand memory (core) for variable constants and working space, and 1K of micro control memory of 48-bit word length. Approximately 7,600 instructions are required with 600 in the executive, 6,000 in the operating programs, and the balance for the test programs.

Differing capacities in processing can be achieved through changes in the number of PROM program memory cards (two required for the basic system), data or processing cards (one card for each eight bits in word processing), operand memory cards (one required), and micro control PROM cards (one required).

## 2.2 RECOMMENDATIONS

To provide the Army with a highly versatile forward area air defense weapon system, it is recommended that the following tasks be performed.

A. Prototype Development

It is recommended that a prototype demonstration AFAADS gunfire system be developed as soon as possible, preferably concurrent with the fulfillment of the specific tasks listed below. Specific answers to certain of the listed tasks are dependent upon the specific sensor, gun, and system configuration used in AFAADS. Satisfactory, unequivocal results can thus be obtained sooner and at less system cost.

The additional specific tasks are in the categories of analysis, software, and hardware.

B. Analysis

1. Target Tracking. Select the optimum technique based upon simulation and testing against real flight data (see FACT test data discussed in Vol. I, Sec. 4). Candidates include fixed length polynomials, recursive polynomials, and adaptive Kalman filters.
2. Ballistic Computation. Optimize the analytic approximations to the ballistic tables for the selected gun. Approximations include polynomials, exponentials, integration of the ballistic equations, etc.
3. Projectile Detection Techniques. Investigate specific techniques for measuring the projectile miss distance at the target for various sensors - radar, FLIR, and laser.
4. Target Acquisition. Optimum techniques for target acquisition by the AFAADS fire unit. Included in this are target designation by other elements and identification procedures and techniques.
5. Data Sampling Frequencies. Analytic determination of the optimum data sampling frequencies between the computer and its peripheral devices, i. e., input/output frequencies.
6. Trigger Control. Analytic determination of the probability of hit as a function of target range, crossing angular rates, cross-section, etc., and hence a determination of the optimum firing rates. Inhibit fire signals could also

be generated if the computer senses that the gun servos can not follow the target accurately enough.

C. Software

1. Flow Chart. Develop detailed flow charts for each of the several computer functions.
2. Timing. The semi-automatic fire control mode is a real time mode. Accurate timing estimates are required for firming up the computer's cycle time.
3. Memory Size. Each of the programs should be carefully sized as to program, operand, and microcontrol memory requirements.
4. Microcoding. Decision on extent to which microcoding should be used rather than minicoding.

D. Hardware

1. Mechanical. Considerations should be given to mounting the computer on the weapon system such that the front surface can be used as the operator's panel.
2. Operator's Panel. Careful layout design with due consideration to human factor requirements. Also the problem of integrating non-computer functions such as the fire switch on the panel should be investigated.
3. Electrical. Selection of circuits, card connectors, wiring, etc.

## SECTION 3

### OPERATIONAL DESCRIPTION

To provide a tactical and operational basis for developing the computer concepts for AFAADS, this section presents brief summaries of:

- a. The operation missions of an AFAADS system
- b. An AFAADS weapon system.

These summaries are primarily restatements of the same topics discussed in greater detail in Volume I of this report and in the two previous AFAADS reports (Refs. 1 and 2).

#### 3.1 TACTICAL EMPLOYMENT OF AFAADS

It is assumed that the AFAADS gun system will fit into the complete Field Army air defense system in a manner that will replace the Vulcan gun and complement the Chaparral missile system. Similar TO and E is also envisioned.

The overall air defense mission relative to a Field Army is to destroy hostile aircraft and missiles, or at least to reduce greatly their effectiveness thereby permitting ease of movement by the ground maneuver elements. The overall air defense is composed of a mix of manned interceptors, gunships, and ground based Air Defense Artillery (ADA) weapons. The ground based ADA weapons complement the interceptors by providing more concentrated defense at key points with a shorter response time.

Within the typical Field Army Air Defense Artillery organization, numerous different weapon types are employed. Each has its own area of specialty. High altitude point defense is currently provided by Nike Hercules. At medium and low altitude, the Hawk missile system provides the most effective defense. The SAM-D missile system will probably replace these missiles in the AFAADS time period. Finally, at very low altitudes, and in particular in the very forward areas, the highly mobile short range weapons are the most effective.

AFAADS is one such weapon. It will replace the air defense now provided by Vulcan and complement that now provided by Chaparral and self-propelled Hawk. It will combine high mobility and self-contained operation with accurate fire (through closed loop or projectile tracking) and reasonable

cost Set-up time is negligible with fire-on-the-move a design objective. Minimal radiation will make it difficult to detect. (The particular system described herein is completely passive except for a laser ranging system, which operates only during actual fire. The use of a millimeter radar sensor will essentially introduce no computer changes.)

More specifically, the AFAADS is being designed to provide the Field Army, with particular emphasis on the forward maneuver elements of a Division, with defense against low altitude strafing and bombing attacks, against helicopter and fixed wing gunships, and against tactical surface-to-surface cruise missiles. Typical target elements to be defended by AFAADS include forward command posts, marching columns, assembly areas, critical base areas, and similar selected units and areas.

The exact deployment of AFAADS would be influenced by the coverage provided by the longer range ADA weapons, the interceptors, and the expected enemy capabilities. Currently, automatic weapons, such as AFAADS, are deployed in increments of two fire units with four weapons constituting a minimum defense.

To be compatible with current Army organizations, the AFAADS gun units will probably be organized into Air Defense Battalions. Such a battalion would be attached to Field Army, Corps or Division. A possible organization having 32 fire units is shown in Figure 3-1.

### 3.2 THE AFAADS WEAPON SYSTEM

The chosen AFAADS gun system for the digital computer concepts task of this volume is based upon the 35-mm Oerlikon gun mounted on a self-propelled tracked vehicle. Any other high fire rate, 20-mm to 40-mm gun requires approximately the same fire control processing. An alternate mount is towed. In all cases, each unit is completely self-contained. The assumed gun has a maximum range of about 4,500 meters.

The Oerlikon gun is chosen for the analysis for several reasons: Ballistics and other gun data are available. The gun has adequate range and caliber for the defined very low altitude forward area air defense problem.

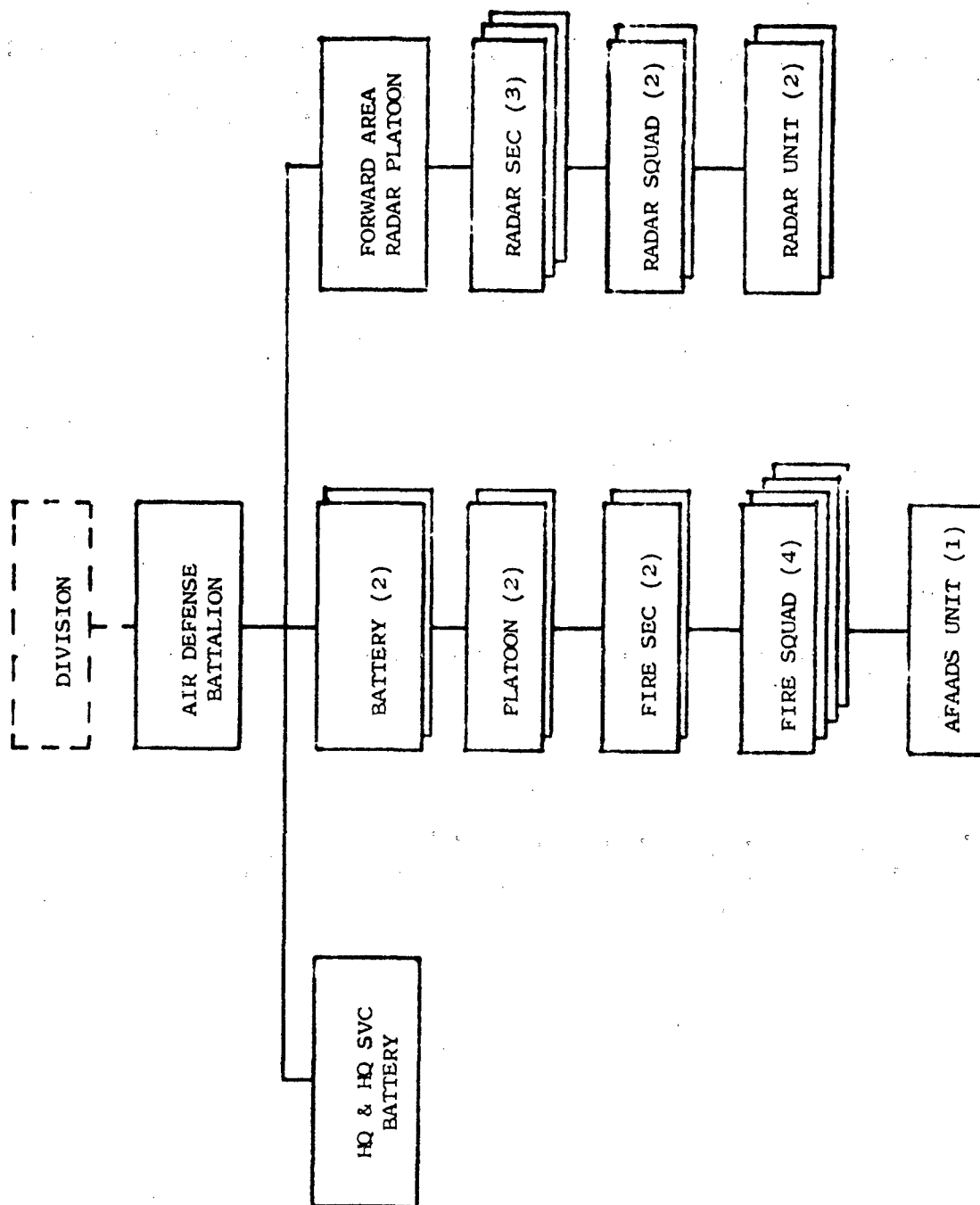


Figure 3-1. AFAADS Air Defense Battalion

This gun is assumed to be completely self-contained with the sensor, computer, power supply, operator and his controls, and magazine mounted with the gun turret on the same vehicle. (See system block diagram, Figure 3-2.) Entire operation of the gun rests with a single Gunner/Operator. The chosen sensor system consists of an acquisition sensor for target acquisition and of a FLIR/laser rate gyro stabilized tracking head for actual fire control. Passivity of this sensing system reduces the the probability of detection and of jamming or spoofing. An alternate sensor is radar, probably in the millimeter band.

For the current analysis, it is assumed that the tracking head is mounted relative to the vehicle, thereby providing it with full 360 degree coverage in azimuth and good elevation coverage (-5 deg. to +85 deg.). The gun turret is mounted relative to the tracking sensor head mount, so its motions in both azimuth and elevation are relative to the sensor head. (Hence the gun is supplied with lead angles relative to the sensor rather than directional angles.) These and other basic AFAADS parameters are summarized in Table 3-I.

In normal semi-automatic fire:

- a) The vehicle is stationary; i. e., no firing-on-the-move, although it is a growth objective.
- b) Target designation (if provided is by voice radio to the operator. Data link reception is another growth capability.
- c) Initial target detection is by the operator using the acquisition system, for instance a helmet mounted sight.
- d) During target search, the FLIR/laser tracking head is slaved to the acquisition system with the laser in standby.
- e) Upon FLIR target detection, automatic angle tracking commences.
- f) Angle tracking activates the laser, followed by target detection and tracking.
- g) The computer develops the fire control solution and delivers lead angle orders to the gun.
- h) Gun opening fire remains an operator function.
- i) Projectile tracking provides a means for bias error corrections.

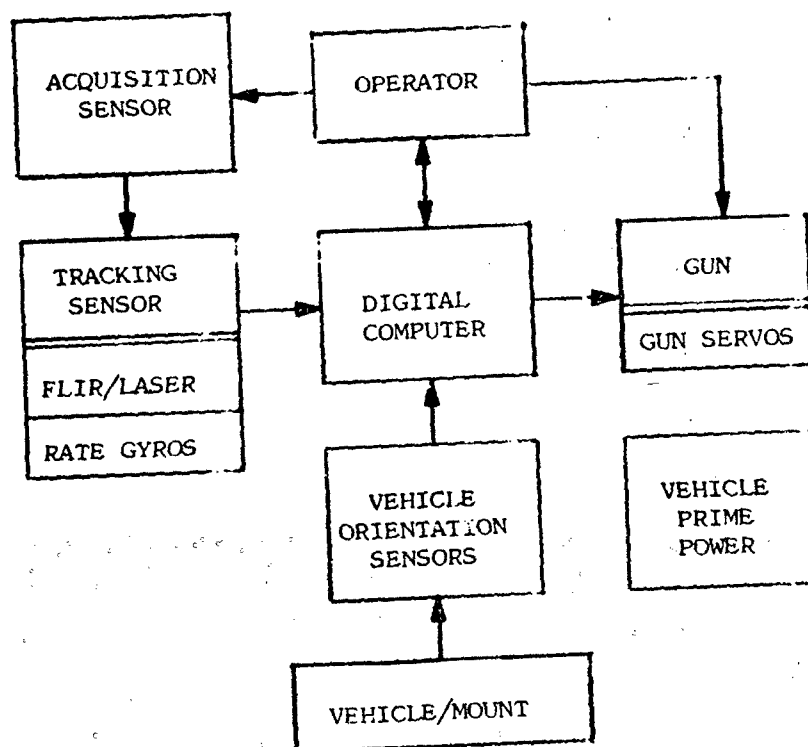


Figure 3-2. AFAADS System Block Diagram

TABLE 3-I Principle AFAAD System Parameters (Sheet 1 of 2)

<u>UNIT</u>	<u>PARAMETER</u>	<u>VALUE</u>
Gun:	Caliber	35 mm
	Rate of Fire	1,100 rounds/min
	Range	4,500 m
	Ammunition	Armour piercing high explosive
		Tracer
		Subcaliber high velocity
	Fuze	Contact
	Accuracy, Aim	0.25 mil
	Magazine	700 rounds
Sensor:      Search:	Type	To be determined (Possibilities: helmet mounted optical sight, standard optical sight and radar)
	Track:	
	Type	FLIR/laser (alternate: radar)
	Target Detection	Automatic with automatic lock-on
	Field of View	FLIR: Search 20 x 40 deg Track 2 x 4 deg
		Laser: 6 mr
	Stabilization	Rate gyros
	Scan Rate	30 times/sec.
	Accuracy	FLIR: 0.25 mil Laser: 0.25 m

TABLE 3-I Principle AFAAD System Parameters (Sheet 2 of 2)

<u>UNIT</u>	<u>PARAMETER</u>	<u>VALUE</u>
Fire Control:	Type	Digital Computer
	Gunner/Operator	1 (one)
	Operating Modes	Semi-automatic Fire
		Manual Fire
	Capabilities	Standby
		Initialization
		Test
		Degraded
		Predicted Fire
		Closed Loop with Projectile Tracking
Vehicle:		Muzzle Velocity Correction
		Meteorological Corrections
	Type	Self-propelled Tracked (Alternate: Towed)
	Capability	Self-contained

In performing the fire control solution, the AFAADS digital computer makes use of several advances in the fire control state of the art. Target position is predicted ahead based upon current tactics (diving, turning, etc.) Ballistic corrections are entered and made relative to changes in muzzle velocity, meteorological conditions, and projectile type. Vehicle tilt is also considered. During fire, projectile miss distance is measured, thereby providing a measurement of azimuth, elevation, and muzzle velocity biases. These are then corrected out in gun lead angle corrections.

## SECTION 4

### FIRE CONTROL COMPUTER CONCEPTS

The operational capabilities of the just described AFAADS weapon system leads to requirements and concepts for the digital fire control computer. These are summarized in this section under three topics:

- a. Fire Control System Logic (Section 4.1)
- b. Software Design Concepts (Section 4.2)
- c. Hardware Design Concepts (Section 4.3)

Further amplification of these topics can be found in Part III, Analysis of the Fire Control System, which starts with Section 6.

#### 4.1 FIRE CONTROL SYSTEM LOGIC

To perform the AFAADS operational requirements against close support attack aircraft and other weapon systems, five operational modes are defined (Figure 4-1).

##### A. Semi-Automatic Fire Control Mode

The Semi-Automatic Fire Control Mode is the principal operating mode for the system. It is used to direct the guns automatically against any target being tracked. The principal operator-performed or assisted tasks in this mode are target detection, acquisition and identification. Actual gun fire also remains an operator function. Otherwise, all tasks are performed automatically. These include target tracking, ballistic solution, gun lead angle determination and ordering, projectile miss detection and bias error correction.

Operator monitoring and override are provided via the operator's panel, integral with the computer face.

##### B. Manual Mode

As a back-up to the Semi-Automatic Fire Control Mode, a Manual Fire Mode is provided. In this mode, the operator takes full responsibility for the fire control solution, gun pointing, and actual firing. The computer is off-line in Standby.

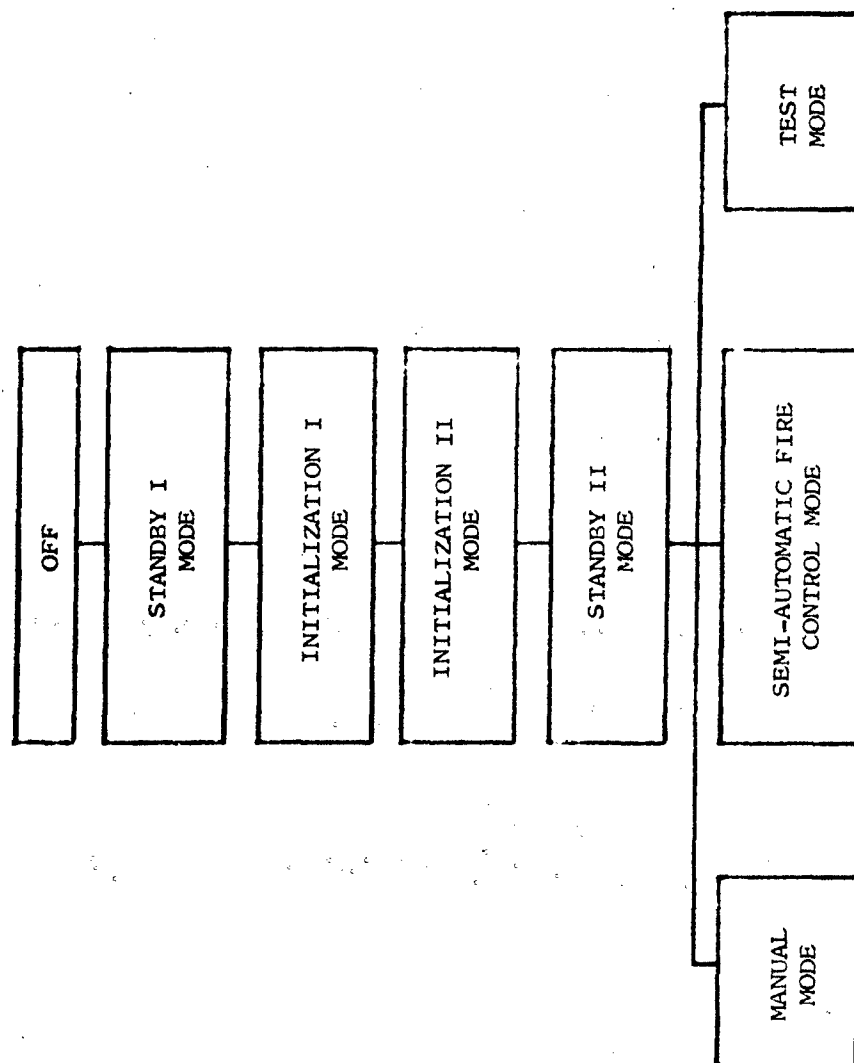


Figure 4-1. AFAADS Operating Modes

Details on this mode have not been pursued since it does not involve the digital computer, the subject of this volume of the final report.

#### C. Initialization Mode

Prior to combat operations, certain parameters must be fed into the computer. For this process, the computer is said to be in the Initialization Mode. Actually, this mode consists of two sub-modes, Initialization I and II. They differ by the types of parameters inputted.

Initialization I deals with slowly varying parameters; not parameters that are apt to change from one mission to the next. These include the meteorological conditions of air temperature and pressure and wind speed and direction. Also, two prediction thresholds must be entered. The predictive fire algorithms in the Semi-Automatic Fire Control Mode include energy conservation during a diving attack and constant turn (acceleration) maneuvering. Others may also be added in the future. Recognition of either of the included ones is by threshold, the values of which must be entered.

Then, Initialization II consisting of the more frequently changing parameters of ammunition type, projectile velocity, and vehicle orientation (i. e., vehicle tilt) is carried out.

#### D. Standby Mode

The computer is in the Standby Mode between fire missions and when not in the Initialization Mode. It is the waiting mode. Like the Initialization Mode, Standby is also in two parts. Standby I is used when the system is first turned on, before Initialization data has been entered. The other part, Standby II, is the normal version. It denotes the computer is waiting for a fire mission. All initialization parameters have been entered.

#### E. Test Mode

In addition to the above four operational modes, a field Test Mode is postulated. This mode provides the Gunner/Operator with

means to test the system in the field. The tests consist of computer confidence tests and an overall dynamic AFAADS weapon system test including live firings and closed loop projectile tracking.

This mode is normally entered from the Standby II Mode.

The operational capabilities being provided by the digital computer are best shown by examining the logic and algorithms of the Semi-Automatic Fire Control Mode. This is a real time mode. Processing must be performed sufficiently rapidly to insure good target tracking and accurate ballistic computation even under extreme enemy attack situations.

For real time operations, the computer performs the following functions every 0.1 second, the period between sensor data inputs:

- a. Target tracking, an eleven point polynomial solution.
- b. Impact Point Prediction, using linear constant energy or dive, and/or constant turn prediction of the impact point.
- c. Ballistic Solution, a fifth power polynomial in range.
- d. Ballistic Corrections, corrections to the coefficients in the ballistic equation and/or to the resulting gun lead angles.
- e. Projectile Miss Distance Processing, determination of the projectile miss distance due to bias errors, rather than target maneuvers.
- f. Bias Error Corrections, development of the appropriate bias error correction factors due to muzzle velocity, azimuth and elevation gun bias errors.
- g. Sensor Head Regeneration, signals to the sensors to maintain tracking during target fades.

#### 4.2 SOFTWARE DESIGN CONCEPTS

This logic forms the basis for the computer software design concepts. To carry out the AFAADS logic requirements, the software is divided into three categories:

- a) Executive, which controls the operation of the computer.
- b) Applications, which perform the fire control computations and also generate a simulated target during dynamic system test.
- c) Test, which provides for computer self-testing.

Within the Executive, system operation is controlled by two types of interrupts: timer and fault indication. The timer interrupts are the prime controlling function in that they poll the various input devices (tracking sensors, operator's panel, etc.) for data every 100 milliseconds (0.1 second). Receipt of new data initiates the appropriate Applications programs. Outputs to the gun and sensor servos are keyed to completion of the applicable Applications programs, rather than to the timer. However, the computer cycle time is sufficiently short that the Application program processing for fire control is completed prior to the next timer interrupt.

Self-testing can lead to a fault indication interrupt. Operator warning is generated and the computer goes into a degraded mode program.

Within this general operational framework, current estimates on the size of the software indicate 7,600 instructions are required. This is divided into 600 for the Executive Programs, 6000 for the Fire Control portion of the Application Programs, and 500 for each of the Simulation and Test Programs.

To perform the indicated real time fire control functions between the 100 millisecond timer interrupts requires an average instruction cycle time of 5 microseconds due to the core memory.

Memory requirements for the programs and the various parameters have been estimated. The above instructions require 14,640 16-bit words of storage including a 20% safety factor. This, together with the requirement for 377 words of fixed parameter storage, indicates a program memory requirement of 15,017 words (or 16K). This must be in (programmable) read only memory. Variable parameters and work space storage account for 1714 additional 16-bit words. Thus, a 2K random access memory (RAM) is required. Non-volatile techniques must be employed for the parameter portion of this storage. (Note: Substitution of other processing algorithms such as Kalman filtering and exponential ballistic processing may change the memory requirements.)

#### 4.3 Hardware Design Concepts

Based upon the operational requirements, the fire control logic, and the software, it is concluded that a single unit, stored program mini-computer satisfies the AFAADS air defense requirements. This machine of about 50 lbs. and 0.77 cu. ft. utilizes state-of-the-art medium scale integration (MSI) logic chips in a modular construction. Eight logic cards plus the operator's panel and the power conditioner make up the basic system. The 16-bit word can be changed in 8-bit increments by varying the number of data logic (arithmetic) cards. Two are used in the basic machine. Similarly, the amount of memory can be changed through the addition (or deletion) of other cards.

Memory storage in the basic machine consists of:

- a. Program Memory - 16,834 words of 16 bits each plus two parity bits. This is a programmable read only memory (PROM) during development. For production, a read only memory (ROM) may be substituted. Two logic cards are required.
- b. Operand Memory - 2,048 words of 16 bits each plus two parity bits. This is a non-volatile core memory for variable constants and working storage. A single logic card is used in the basic machine. More can be added.
- c. Micro-Control Memory - 1,024 words of 48 bits of either PROM or ROM. A single card is in the basic machine.

In addition to these five cards and the two arithmetic ones, there is an input/output logic card which includes an interval timer and a mini-control card.

The computer contains two external connectors, one for power and one for data signals with the peripheral devices. There is also an internal test connector for depot maintenance testing of both the hardware and the software. The data signals with the peripheral sensors and servos (gun and sensor mounts) are all 16-bit parallel with digital buffers in the sensors and servos.

The operator's panel is laid out from a functional flow viewpoint. Lines and arrows on the panel surface provide cues to the operator on the desired sequence of operations from initial turn-on through the end of a fire mission or system test.

All operations on the panel are under computer control, rather than hardware control. This provides a high level of flexibility to meet changing requirements during the development phase and reduces the dependence on mechanical components with their normally associated high failure rates. Software interlocks are provided to reduce the likelihood of inadvertent button pushing during critical gun fire operations - such as during semi-automatic fire. Such interlocks also are used to eliminate erroneous code entries, such as would occur with the simultaneous activation of two pushbuttons.

## SECTION 5

### EXAMPLE OF AFAADS OPERATION

This section of the report discusses three different aspects in the operation and employment of the AFAADS gun system used in the digital computer concept studies. These are:

- a) Operating Modes: A summary of the various operating modes being designed into AFAADS computer.
- b) Typical Fire Mission: Trace of the operation of the AFAADS in defense against a high speed air attack on a close-by target.
- c) Operator Controls and Displays: A summary of current concepts on the necessary operator controls and displays for AFAADS.

Each of these three subjects is covered in the following subsections.

#### 5.1 OPERATING MODES

A tactically useful AFAADS system must be provided with several different operating modes. These are required to cover the various operational and tactical situations in which AFAADS must operate. They are also required in order to provide some capability in the face of partial battle damage and/or partial equipment failure. Other situations that must be covered include testing and training. Each of presently defined operating modes are now discussed, starting with the principal air defense mode.

- a) Semi-Automatic Fire Control Mode: The Semi-Automatic Fire Control Mode is the primary air defense mode used in the AFAADS computer design concepts effort. Initial target detection is the operator's responsibility. Lock-on is automatic by the FLIR/laser tracking head. This action initiates computer tracking of the target, target path prediction, and gun ballistic solution. After making appropriate ballistic corrections due to meteorological and other corrective factors, gun lead (i. e., pointing) angles are developed. The gun automatically slews to the

designated direction. The gun position is continually updated based upon updated target position data. Actual gunfire, however, remains with the operator.

With the initial firing, projectile miss distance measurements are made, thereby providing data for gun bias corrections in azimuth, elevation, and muzzle velocity.

Under the design concepts, rapid and accurate computations optimize the response of AFAADS to target maneuvers. Specific target prediction techniques incorporated into the system are linear diving, constant turn, and, as a growth feature, defense of a known point. (Analyses of these techniques can be found in Volume I and the two previous reports - Refs. 1 and 2).

- b) Manual Mode: The Manual Mode provides a degraded mode fire capability in the event the computer and/or one of its peripherals become inoperative. It also provides a capability in the face of countermeasure activity against the FLIR/laser tracking head.

In the Manual Mode, essentially all of the AFAADS equipments are used except for the computer. The operator now has the responsibility for developing the appropriate lead angles for the gun, probably based on data from a conventional optical tracking sight.

Since this mode does not use the digital computer, and the concepts on the latter are the subject of this volume, specific design and operational details have not been worked out.

- c) Initialization Modes: Prior to semi-automatic fire, certain variables must be entered into the computer. These are entered during the Initialization Mode. Actually two such modes are envisioned. Initialization I is used to enter slowly changing variables, ones which may change only two or three times a day. Specific entries are meteorological data of air temperature and density, wind velocity and direction. Other entries are the thresholds to be used in connection with the various prediction techniques. If such data are unavailable, standard or preset values, stored in the computer, are used.

Then there are those parameters which may vary with the particular fire mission. They make up the Initialization II Mode. Ammunition type, vehicle orientation, and expected muzzle velocity lie in this category.

- d) Standby Modes: The present design concepts are for two Standby Modes. Standby I is the mode assumed by the computer after turn-on. At this time no Initialization I data has been entered. After Initialization I data has been entered, the computer goes to Standby II. This latter is also the ordered mode entered at the end of a fire mission (unless the system is shut down). In Standby II, the computer and other AFAADS elements are ready for mission assignment.

When the Semi-Automatic Fire Mode is entered, the system first goes to Initialization II to read in the appropriate parameters. Then, it immediately goes into the designated fire mode.

- e) Test Mode: The AFAADS computer concepts consider field system tests. The simplest provides a logic check that the computer is operating correctly.

The most complex is an overall AFAADS dynamic system test. Actual ammunition is fired for calibration purposes against a simulated target. Miss distance measurements relative to the simulated target position provide bias correction data. Such a system test might be employed daily or following movement to each new defensive position.

- f) Degraded Modes: Failure of certain elements in the AFAADS system need not abort the mission. One example of a degraded mode is the Manual Mode described above. This covers loss of the digital equipment. Other degraded modes include loss of the FLIR or of the laser. Alternate operations could include employment of a standard optical sight with range estimation by the operator. This area needs further exploration.
- g) Ground Fire Mode: Although not considered in the current concept studies, due consideration should be given to fire against surface targets.

## 5.2 TYPICAL FIRE MISSION

To illustrate the operation of the AFAADS gun, particularly the digital fire control computer, we now consider the principal events during the defense against a high speed low altitude aircraft attack. To be specific we consider the AFAADS gun defending a point target (command post, bridge, etc.) located 200m due north of the gun. The gun mount is also assumed to be facing north. The terrain is flat.

A near sonic (300m/sec) jet aircraft makes a straight and level attack run from the East against this target, passing over at 200m altitude. (Figure 5-1 shows the defense geometry.)

The resulting air defense operations performed by the AFAADS gun are illustrated in Figure 5-2 in a time event plot. In the figure, time is measured in terms of time-to-go before the point of closest approach. Hence, time becomes negative as the aircraft departs. The range between the aircraft and AFAADS is also plotted. At the point of closest approach, this becomes 283m (200m ground range and 200m altitude). The principal events are annotated below the time and range lines.

Prior to target approach, the AFAADS gun has been initialized through the input of slowly varying parameters, primarily meteorological data. Initial target warning is obtained by voice radio to the operator. He then activates the system into the Semi-Automatic Fire Control Mode. This causes the gun mount and the tracking sensor to be slaved to the acquisition system. Upon target detection and lock-on by the assumed FLIR/laser tracking sensor, computer tracking commences and the fire control solution is obtained. Appropriate lead angles are generated and fed to the gun mount.

When the target comes in range, firing commences. This is approximately 22 seconds before the time of closest approach, and at 8300m range. This initial fire is open loop, since seven seconds are required for the first projectiles to reach the target (at 4500m range). At this time, 15 seconds from closest approach, the first projectile miss measurements are made. Bias corrections are immediately made and corrective rounds fired. This process continues for the next few seconds when the sensor slewing rates become too great for accurate miss distance measurements (see Appendix D).

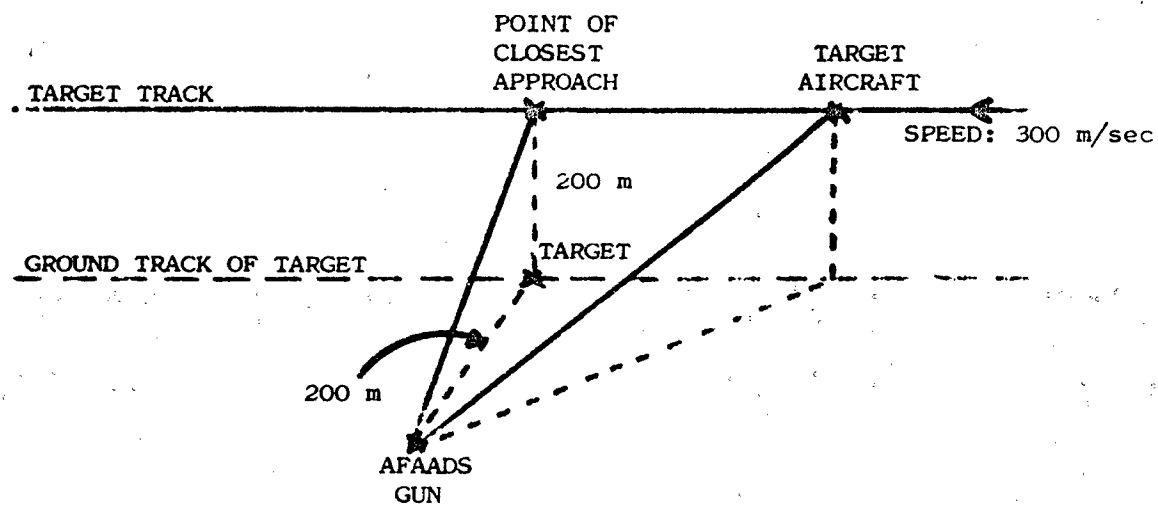
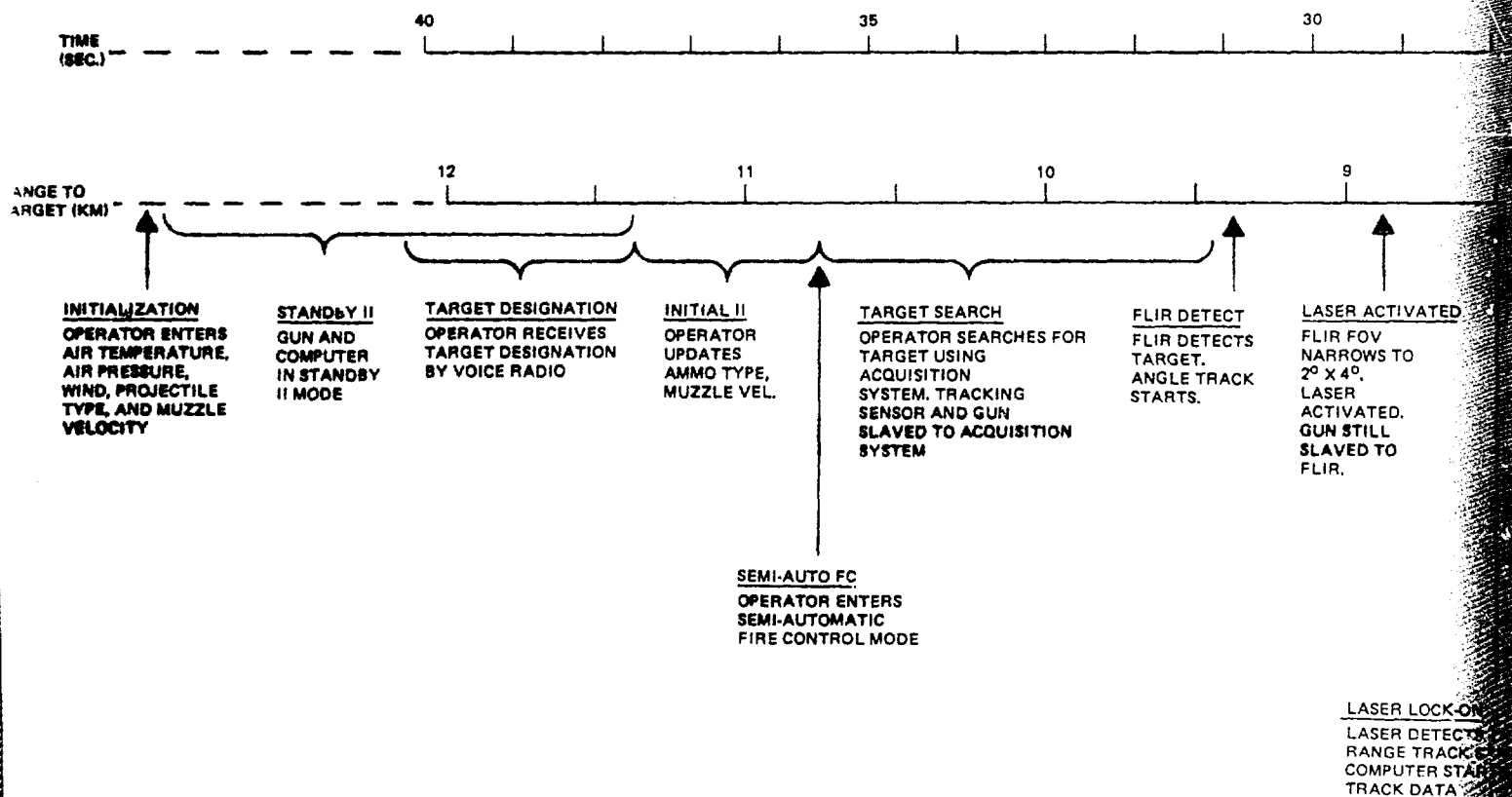


Figure 5-1. AFAADS Engagement Geometry



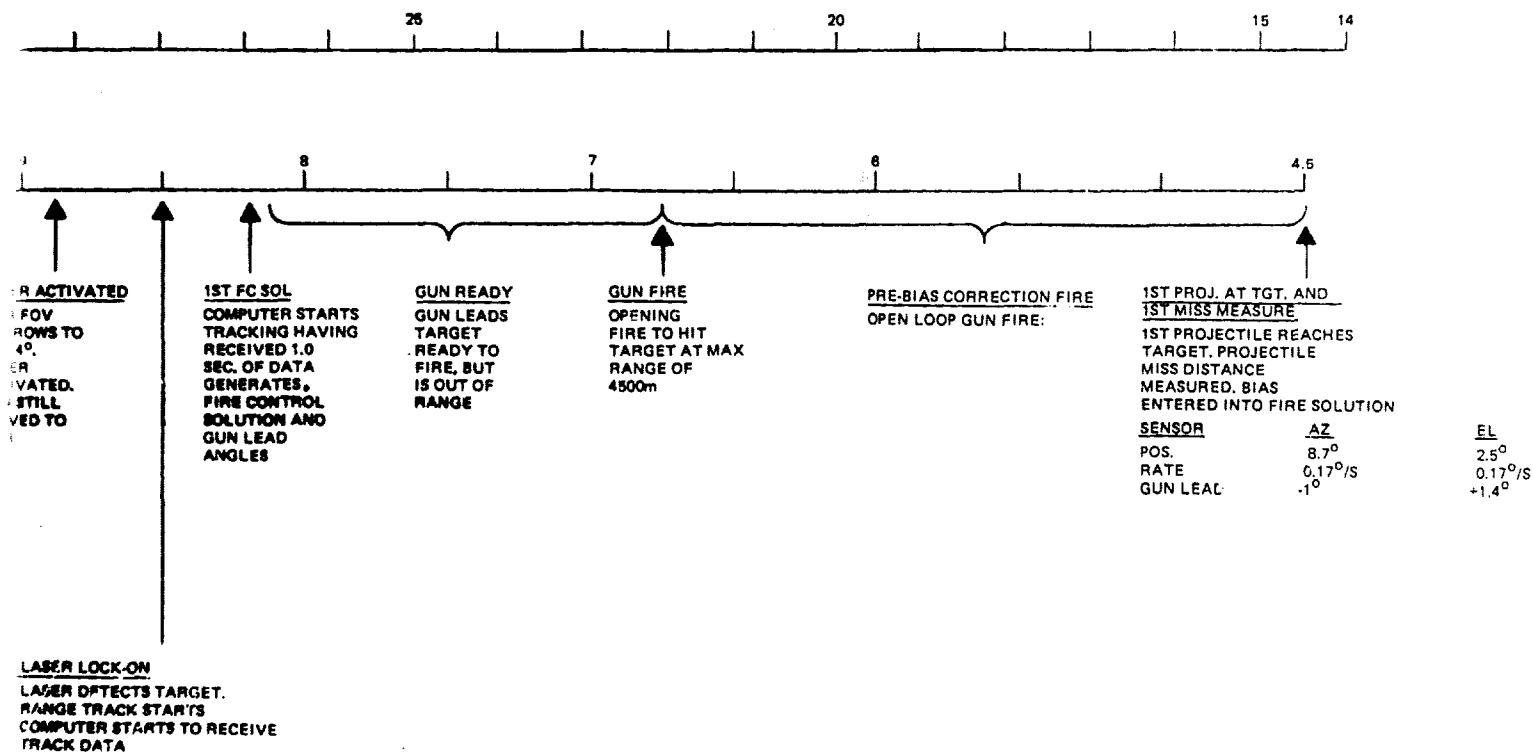
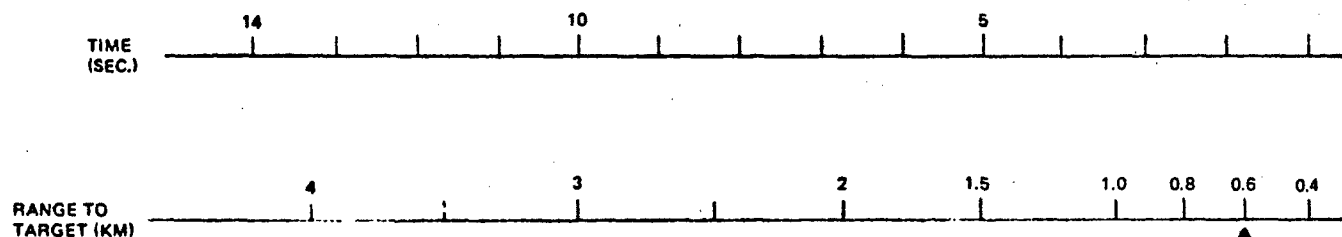


Figure 5-2. Typical AFAADS Fire Mission Time Event Plot.



**PROJECTILE MISS CORRECTIONS**  
 PROJECTILE BIAS MISS  
 CORRECTIONS MADE  
 DURING THIS 4 SEC.  
 PERIOD

LAST PROJECTILE  
 MISS MEASUREMENTS  
 ANGULAR RATE EXCEEDS  
 0.28°/SEC. MAX RATE  
 FOR PROJECTILE MISS  
 MEASUREMENTS (SEE  
 APP. D.)

**CRITICAL TR**  
 SENSOR SLE  
 10°/SEC. AND  
 ACCELERATI  
 LEAD ANGLE  
 20.4° IN AZ.

#	t	D	PC
			AZ
1	1.8	610	69.3°
2	0.25	293	20.4
3	0.0	283	0.0
4	-1.8	610	-69.3



The most interesting and also the most critical part of the firing run occurs 1.8 seconds on either side of the point-of-closest approach. It is in this region where the sensor slewing rates become very large, peaking at 85.5 deg per sec in azimuth. The greatest required lead angles of the gun relative to the FLIR/laser tracking head are also present. Thus servo limitations in slewing rates for either or both the sensors and the guns can cause appreciable angular misses - just at the time where the range is a minimum and hence the probability of kill the greatest. Should the gun "get behind" where it ought to be, a period of ineffective fire will occur until the weapon catches up. Computer generated leading signals can reduce this effect appreciably. By then the target is departing. Cease fire will occur approximately seven seconds after this point of closest approach.

To see the exact variation of these parameters with time, reference is made to Figures 5-3, 5-4, and 5-5. These deal with the three polar coordinates of range to target, target azimuth or bearing measured from the point of closest approach, and target elevation. The position, rate, and acceleration values are all plotted as a function of time from the point of closest approach.

### 5.3 OPERATOR CONTROL AND DISPLAYS

An important part of the AFAADS computer design concepts is the man-machine interface through the Operator's Panel for the computer. This panel, illustrated in Figure 5-6, will be mounted directly on the front of the computer, thereby eliminating additional cabling and connectors.

The functions performed by the Operator's Panel are three-fold:

- a) Fire Control - providing a means for positive control over the operation of AFAADS during an air defense mission.
- b) Status - providing the operator with indications on the operation of the digital computer and the fire control system.
- c) Data Entry - providing means for the operator to enter data into the computer. These data include both fire control and initialization.

Details on individual button operations and indicator functions are presented in Section 7.

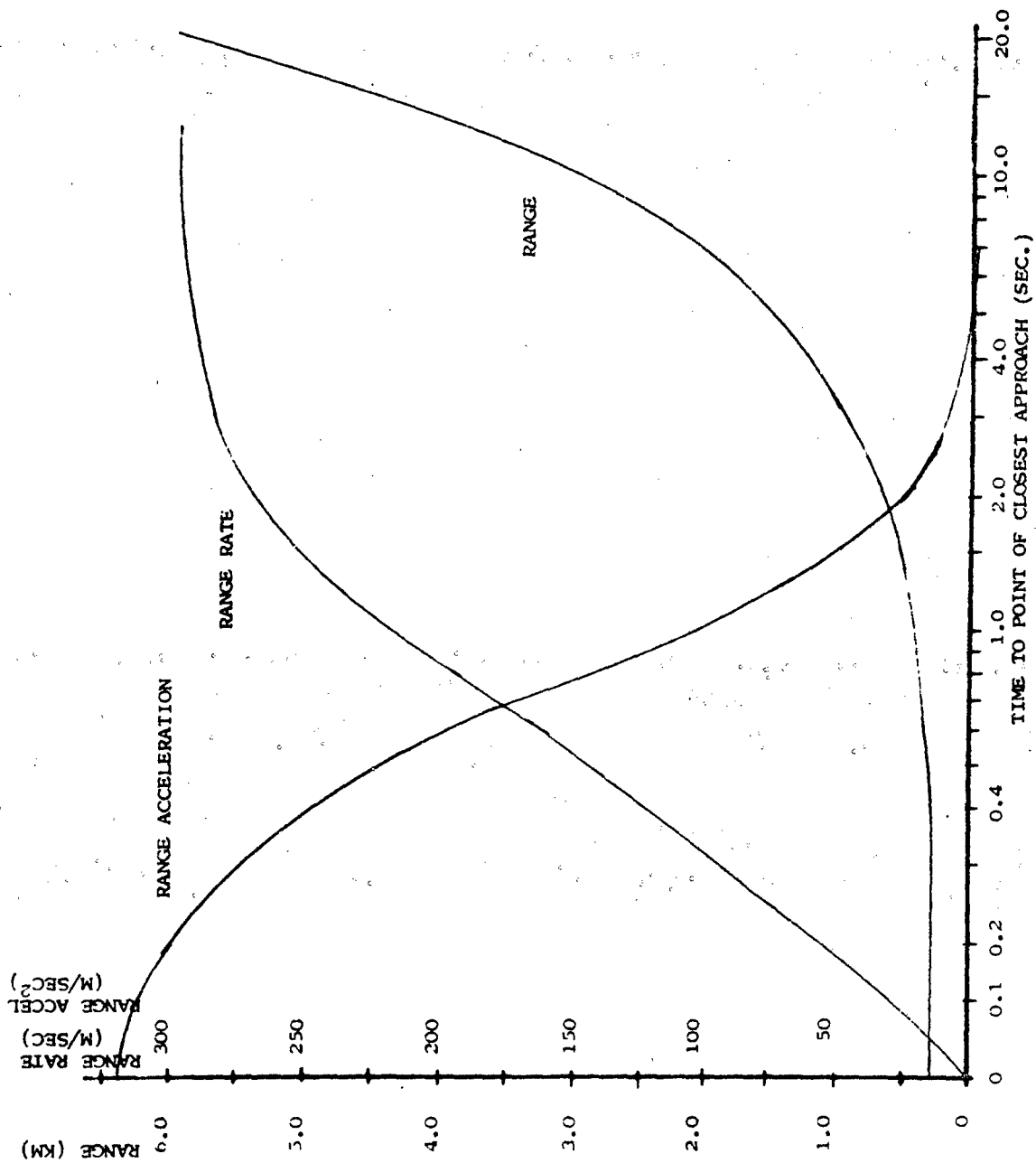


Figure 5-3. Target Range from AFAADS Gun

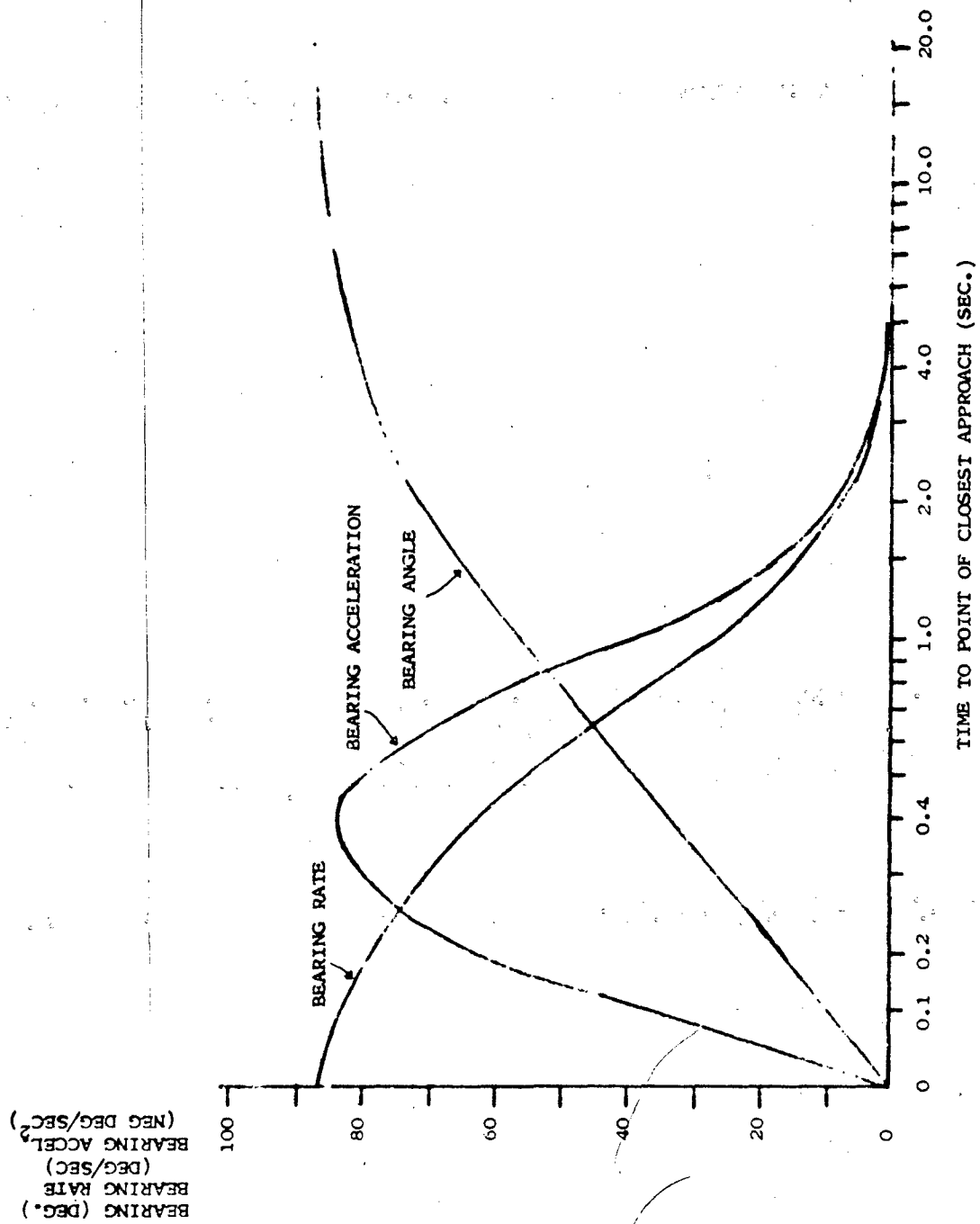


Figure 5-4. Target Azimuth from AFAADS Gun

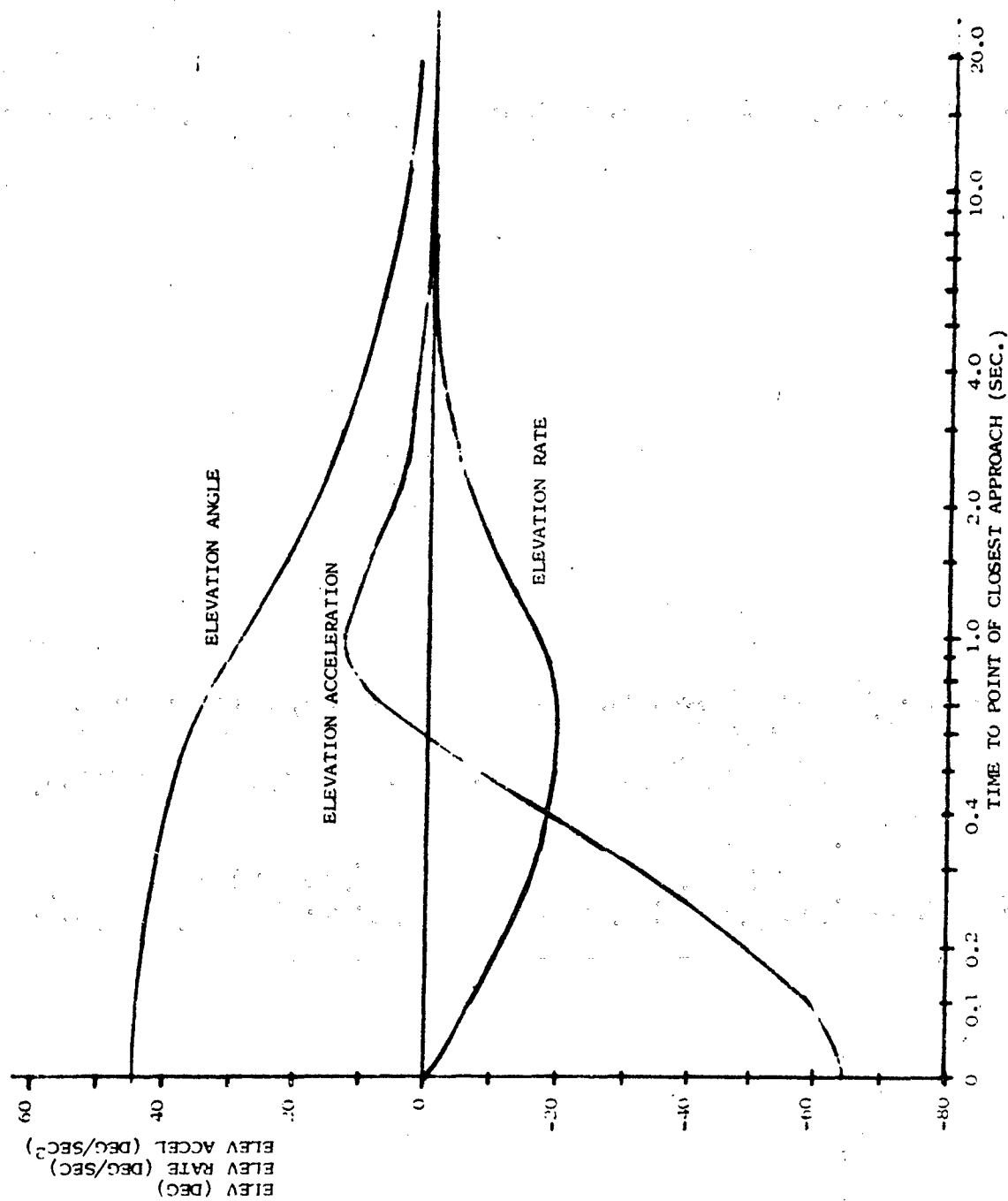
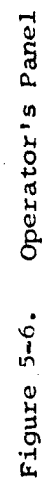


Figure 5-5. Target Elevation from AFAADS Gun



The Operator's Panel as shown in Figure 5-6 contains all three functions. Additional controls and displays will be required to cover functions related to other AFAADS elements besides the digital computer, for instance the firing switch. (This is one area recommended for analysis in a follow-on program.)

### PART III. ANALYSIS OF THE FIRE CONTROL SYSTEM

The remainder of this volume, consisting of Sections 6 through 13, presents the detailed analyses performed on the AFAADS fire control system in developing the digital computer concepts and sizing. The work begins with an overall system synthesis for meeting the operational requirements (Section 6). Also included in this section are the basic parameters of the assumed weapon and sensors. This is followed in Section 7 by a discussion of the Operator's functions, his displays and controls. Typical attack scenarios is the presentation medium used. With this discussion, the computer-Operator interface is well-defined from an operational viewpoint. Section 8 carries this one step further with an analysis of all the signal interfaces with the computer; the Operator's panel, the sensors, and the gun and sensor servos.

With this analytic basis, the required AFAADS software or computer programs form the basis of Section 9. This is followed in Section 10 with a presentation of the computer hardware concepts. The critical requirements met by these concepts are largely those resulting from the real time requirements of semi-automatic fire control. For this reason, a complete set of processing algorithms and logic are developed in Section 11 for the real time operations. Other operating modes are necessary. Two are more fully developed. Section 12 deals with certain degraded modes and Section 13 with test modes employed by the Operator in a forward area.

## SECTION 6

### SYSTEMS ANALYSIS AND OPERATIONAL REQUIREMENTS

The AFAADS gun defense system is required to operate in forward areas in defense of point targets. High mobility and self-contained operations with rapid set-up and knockdown times are clearly required.

These operational requirements develop into overall system requirements, the topic of Subsection 6.1. These system requirements result in the definition of specific operating modes. From the modes, top level computer logic and flow charts evolve (Subsection 6.2). This section then concludes with a listing of the assumed basic AFAADS parameters used in the computer concepts analysis (Subsection 6.3).

As previously noted, a set of parameters is assumed, in order to have a tractable problem for computer sizing. Other similar parameters would, it is believed, lead to essentially the same size and concepts for the processor.

#### 6.1 BASIC FIRE CONTROL SITUATIONS

Let us consider the employment of an AFAADS gun system in a typical forward area environment. A typical point target being defended might be a command post, say brigade or division headquarters, or a field artillery fire direction center. Other targets include an artillery battery, an advancing column, supply columns, or a bridge (captured or pontoon) or crossroads.

In each case, the target to be defended occupies a relatively small geographic area. It is often a temporary installation. The column moves on, the headquarters is echeloned forward, the area is made more secure.

Thus, an AFAADS gun system must be able to move in rapidly, set-up and provide significant air defense in a short period. Redeployment to another target point must also be rapid. AFAADS cannot rely upon an elaborate air defense network. It must be fully self-contained.

A basic AFAADS design requirement evolves from this. All elements of the gun weapon system must be contained on the same vehicle. This includes the sensors, fire control computer, gun, ammunition magazine, Operator's control station, and the prime power source. A tracked vehicle is an ideal mount. An alternate is a towed mount.

Thus, when AFAADS reaches an assigned target area, the vehicle is stopped. Dust covers are removed. Prime power is applied and the system should be ready to operate. Omitted in this concept are such time-consuming tasks as deploying and sighting of sensors, laying and hooking-up cables.

To further speed the set-up time, all initialization parameters are, when once entered, permanently stored in the computer until changed by the Operator. Thus, meteorological data can be entered every six or twelve hours and remain available independent of how often the AFAADS vehicle is moved to another defensive position.

These two AFAADS concepts also provide the basis for later expansion to a fire-on-the-move capability.

Let us now look at the actual defense situation itself. A close air support jet aircraft can approach the defended point from any direction and use numerous different attack profiles. As the aircraft passes over the target area, the angular tracking rates and accelerations required of AFAADS can become very high (see Ref. 1). Closed loop fast response servo tracking is the only possibility of keeping on target. This means automatic tracking, fire control computation and gun slewing. These are basic to the AFAADS concepts. Tests (see FACT data in Volume I, Section 4) show certain characteristics in attack aircraft trajectories. Certain of these are incorporated into the AFAADS concepts; particularly total energy conservation experienced in a diving attack and the acceleration in a constant turn attack. These are a part of the target prediction function used in connection with the ballistic computations.

Now a tracked vehicle going over rough terrain is subjected to all types of stresses and bending moments. Precise alignments between gun and sensor will be difficult to achieve, let alone maintain. For this and other reasons, the AFAADS incorporates closed loop or projectile miss measurements. These are converted into ballistic bias corrections in azimuth and elevation. A rapid fire high velocity weapon will also experience considerable tube wear; thereby affecting the muzzle velocity. Both operator-entered and bias error corrections to this problem are provided.

The air defense provided by AFAADS will not always be against high speed jet aircraft. Helicopter and fixed-wing gunships and cruise missiles may also be involved. Occasionally, the operational situation may require fire against ground targets. Thus, different types of ammunition should be available, along with the appropriate ballistic solutions. This means a requirement for operator selection of ammunition type.

We have now defined two basic operating modes; an Initialization Mode, when the several variable parameters are entered, and a Semi-Automatic Fire Control Mode, when the target is tracked and fired upon. The required response time does not permit the entry of initialization data after a target is detected. This defines the need for an intermediate mode, Standby. In Standby, the system is up and ready. Target engagement can proceed immediately.

Unfortunately, equipment failures can always exist. Also, target signatures may not be as clear as desired; i.e., noisy sensor data. Countermeasures may be employed. Thus, a back-up Manual Mode capability is required, a capability for the Operator to personally direct the gun in fire.

Any Gunner will want to be sure that his equipment is operating. Thus, a Test Mode is required. Ideally, this should cover both the individual elements in the systems and the entire system. Both of these have been levied as requirements on AFAADS.

In summary, we have defined five operating modes and certain characteristics for each. They are:

1. Initialization Mode: Variable parameters can be entered. In addition, provision is made to retain the last used values during shut down to speed up equipment set-up time. Also, since all elements of the AFAADS are on one vehicle, no siting parameters are required. Actually, it is found convenient to divide this into Initialization I for the slowly varying parameters and Initialization II for those parameters that are changed more often.
2. Standby Mode: This is the waiting mode. It has also been subdivided into Standby I, before the entry of initialization data, and Standby II after this data has been entered.
3. Semi-Automatic Fire Control Mode: This is the principal fire control mode. It includes automatic fire solution, target prediction based upon expected target trajectories, and closed loop (projectile miss) tracking. Actual gun fire is reserved as an Operator function.
4. Manual Mode. A manual back-up mode.
5. Test Mode. Forward area battlefield tests of the AFAADS elements and the complete system.

## 6.2 COMPUTER LOGIC STRUCTURE

Once the above five operating modes have been defined, a logic structure evolves between them. In a normal operation, when the AFAADS gun system reaches its assigned defensive position, the equipment is turned on (Figure 6-1). It automatically goes (defaults) into the Standby I Mode. The Gunner/Operator will often wish to review and/or enter new initialization data. For this purpose, he places the system in the Initialization I Mode. Upon completion of data entry, the system is placed in Standby II Mode awaiting an attack. Normally, the weapon fire would be conducted in the Semi-Automatic Fire Control Mode. Under degraded conditions, the Manual Mode could be used. If time is available, system check-out would be carried out in the Test Mode. Completion of a fire mission would normally see the AFAADS returned to Standby II.

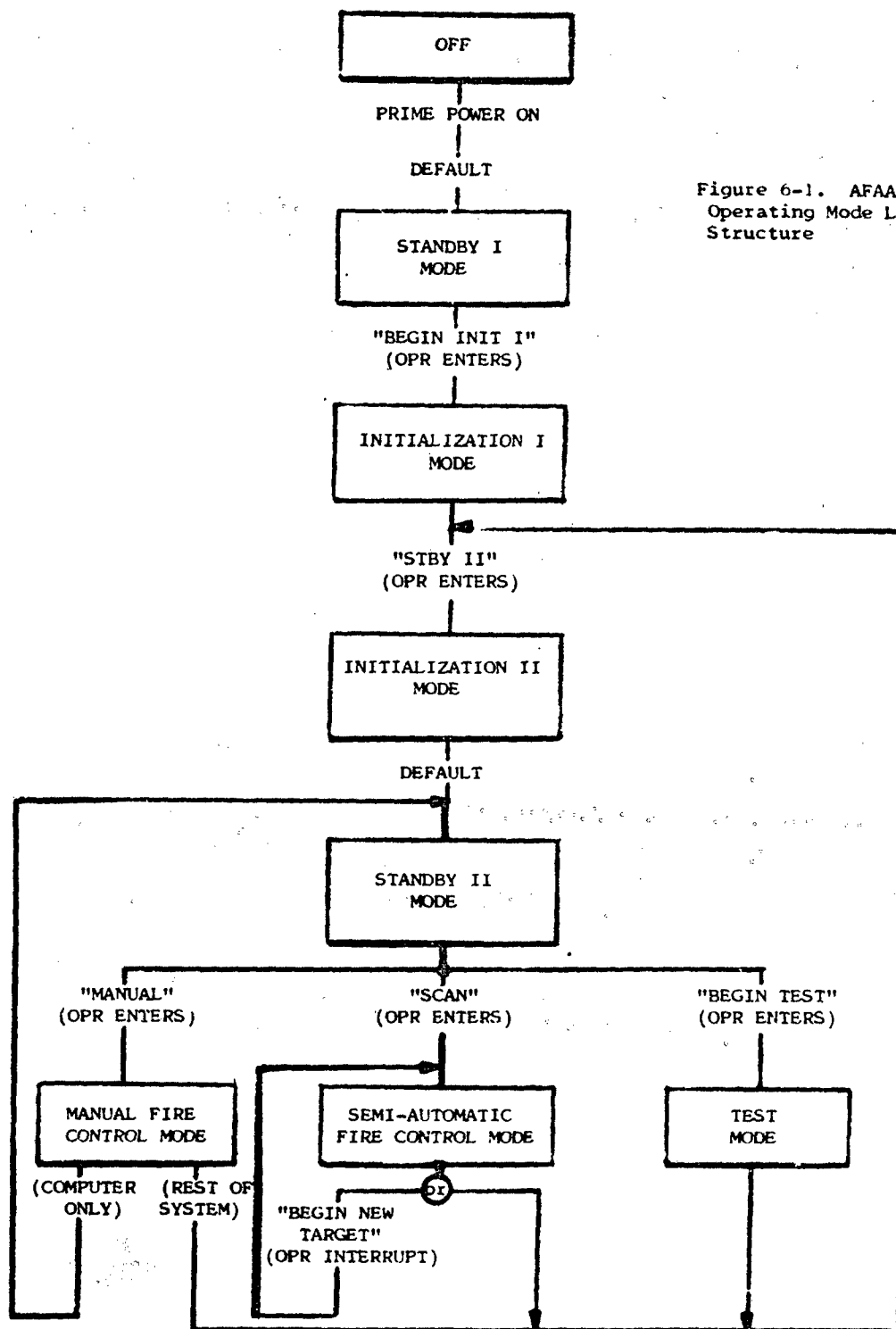


Figure 6-1. AFAADS  
Operating Mode Logic  
Structure

It should be noted that the sequence of Figure 6-1 need not always be followed. The system is designed so it can go from any one mode to essentially any other mode. For instance, new initialization data can be entered after a fire mission, or the Test Mode can go directly to Semi-Automatic Fire if an enemy target suddenly appears. Any other combination is also possible.

The operating mode logic structure of Figure 6-1 has been expanded in Figure 6-2 to depict the more detailed sequence of events in the normal operating situation. That is, the order in which the various modes are passed through is the same on the two figures. On Figure 6-2, any phrase or word contained in quotes designates an indicator on the Operator's Display which is lit for some period of time. Thus, "Scanning" signifies SCAN on the Operator's Display is lit steadily for some period of time.

The Operating Situations Flowchart, Figure 6-2, begins with an initialization of parameters (Initialization I Mode) such as air temperature, wind direction, etc., that are infrequently changed by the Operator (see Figure 5-6, Operator's Display). A second initialization (Initialization II Mode), done automatically and immediately preceding going to Standby II Mode, results in a scanning of the Operator's panel for manually entered initial muzzle velocity and type of round, and of the vehicle orientation. These variables may change engagement-to-engagement, or test-to-test. During manual use of the system (Manual Mode) where all operations are done by the Gunner-Operator, neither initialization is used, but the data remains in the computer, should a quick change of modes be initiated.

During this "Manual" use of the system, the tracking sensors would supply data directly to the operator. The gun would be slaved to an operator control outside the computer. While in the Manual Mode, the computer is bypassed. The computer waits in Standby II Mode, a standby-with-data mode, indicating readiness for immediate use if desired.

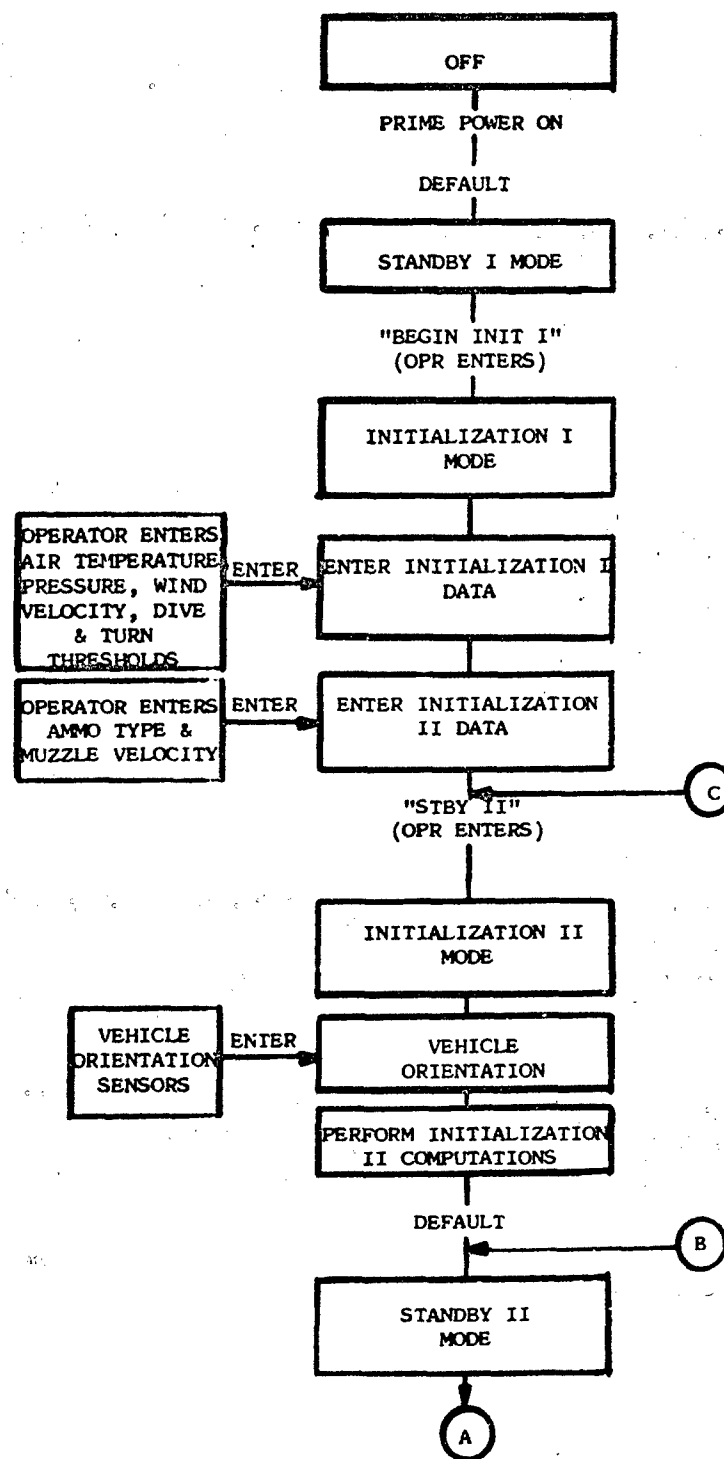


Figure 6-2. Operating Situations Flow Chart (Sheet 1 of 5)

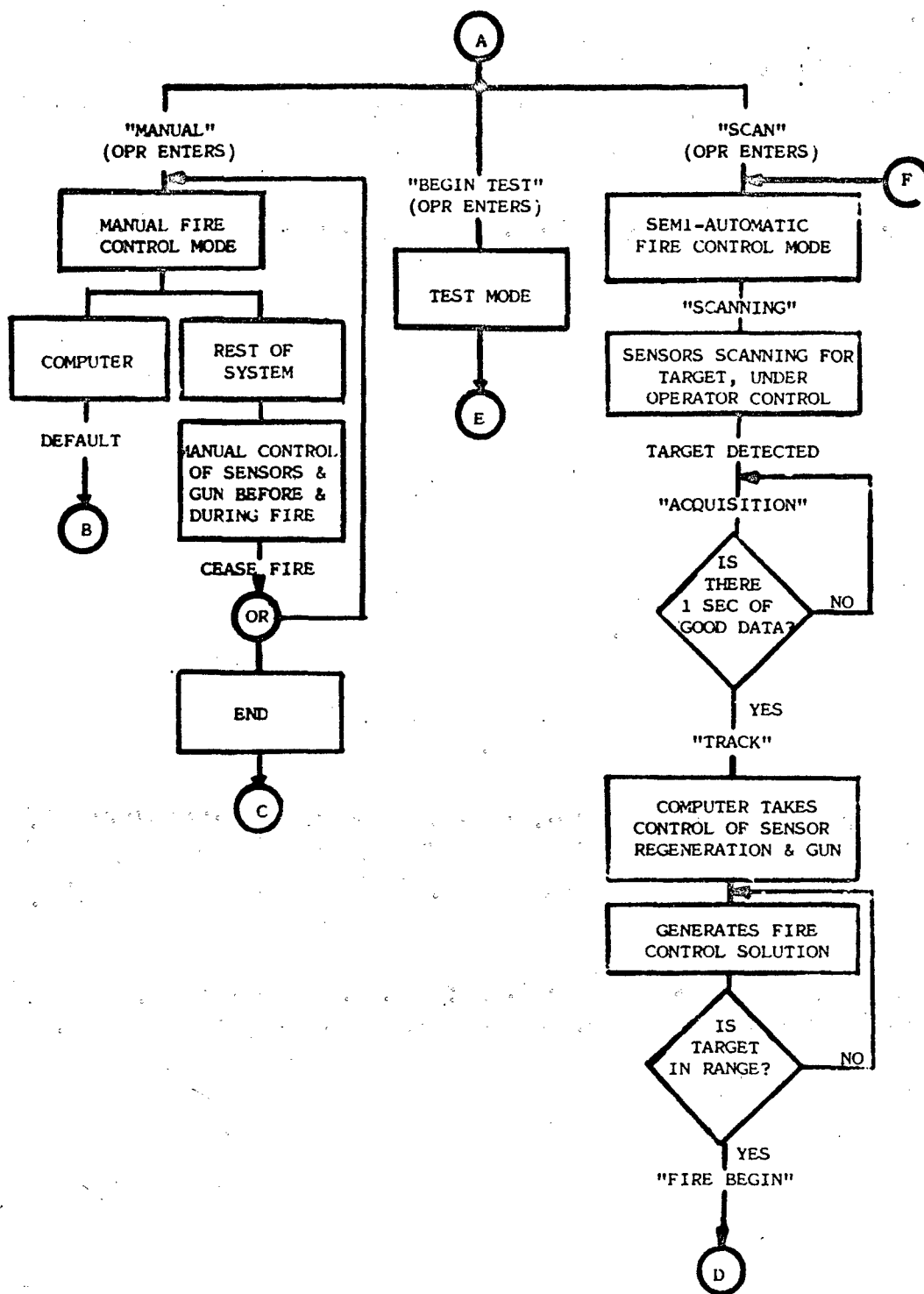


Figure 6-2. Operating Situations Flow Chart (Sheet 2 of 5)

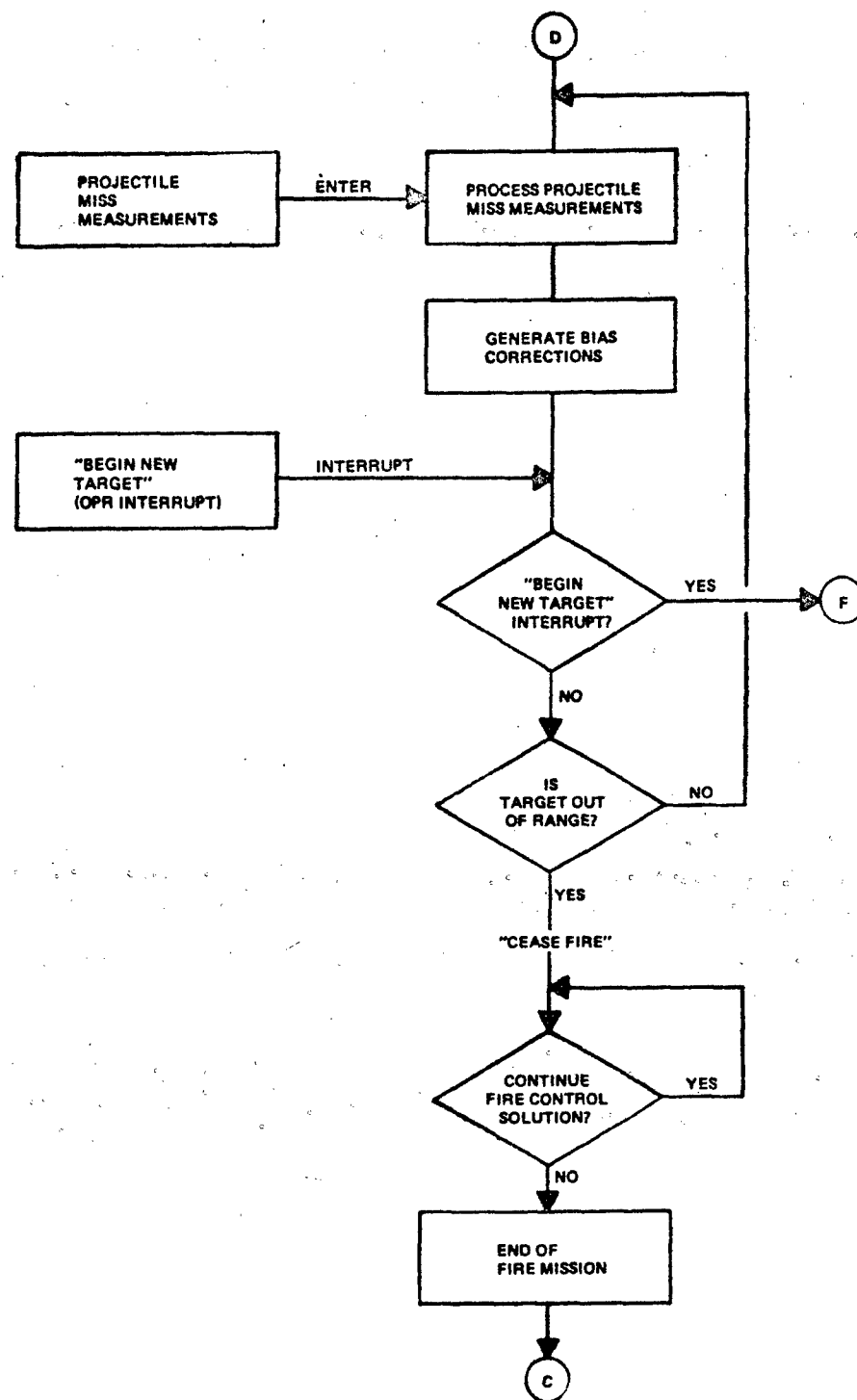


Figure 6-2. Operating Situations Flow Chart (Sheet 3 of 5).

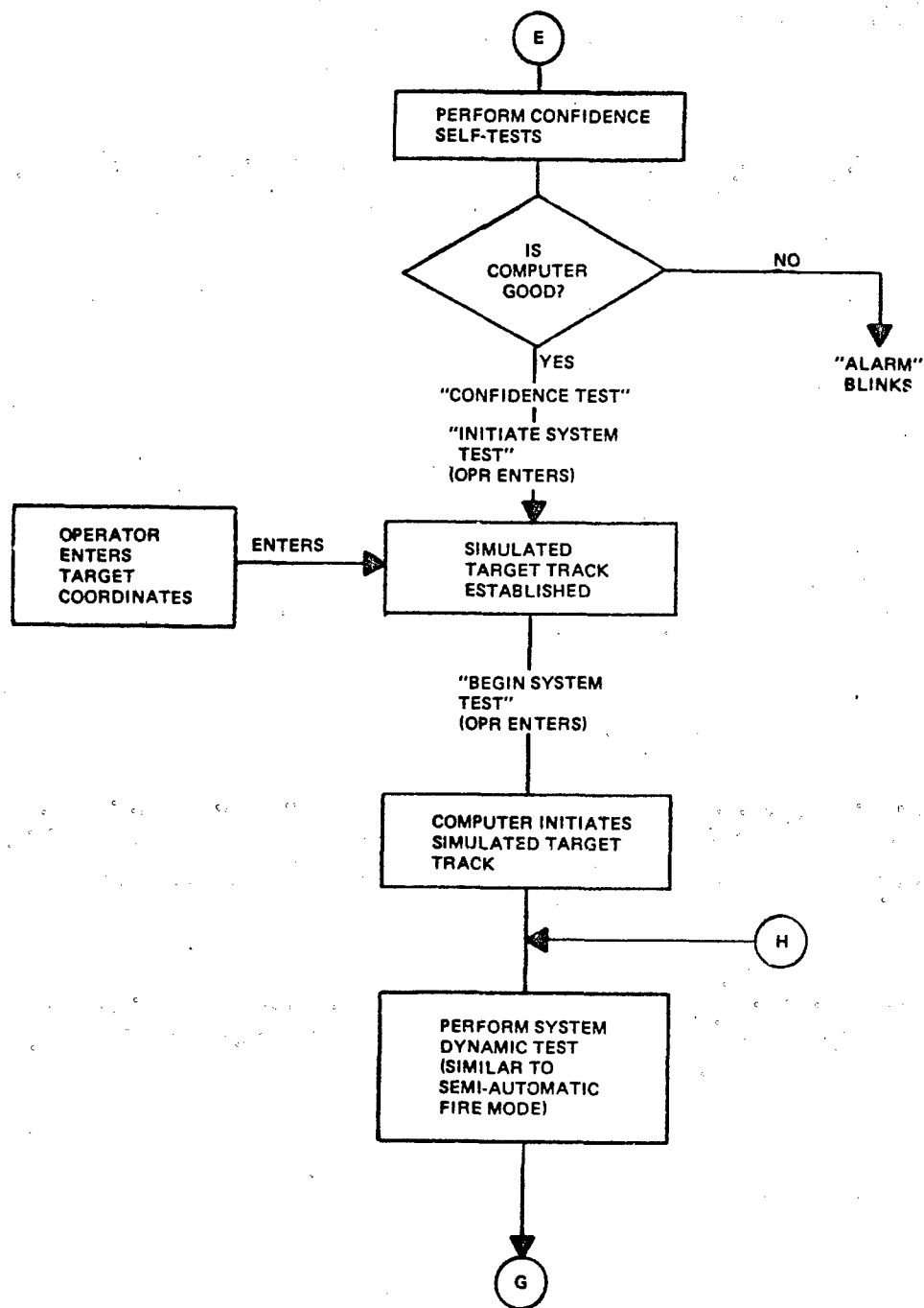


Figure 6.2. Operating Situations Flow Chart (Sheet 4 of 5).

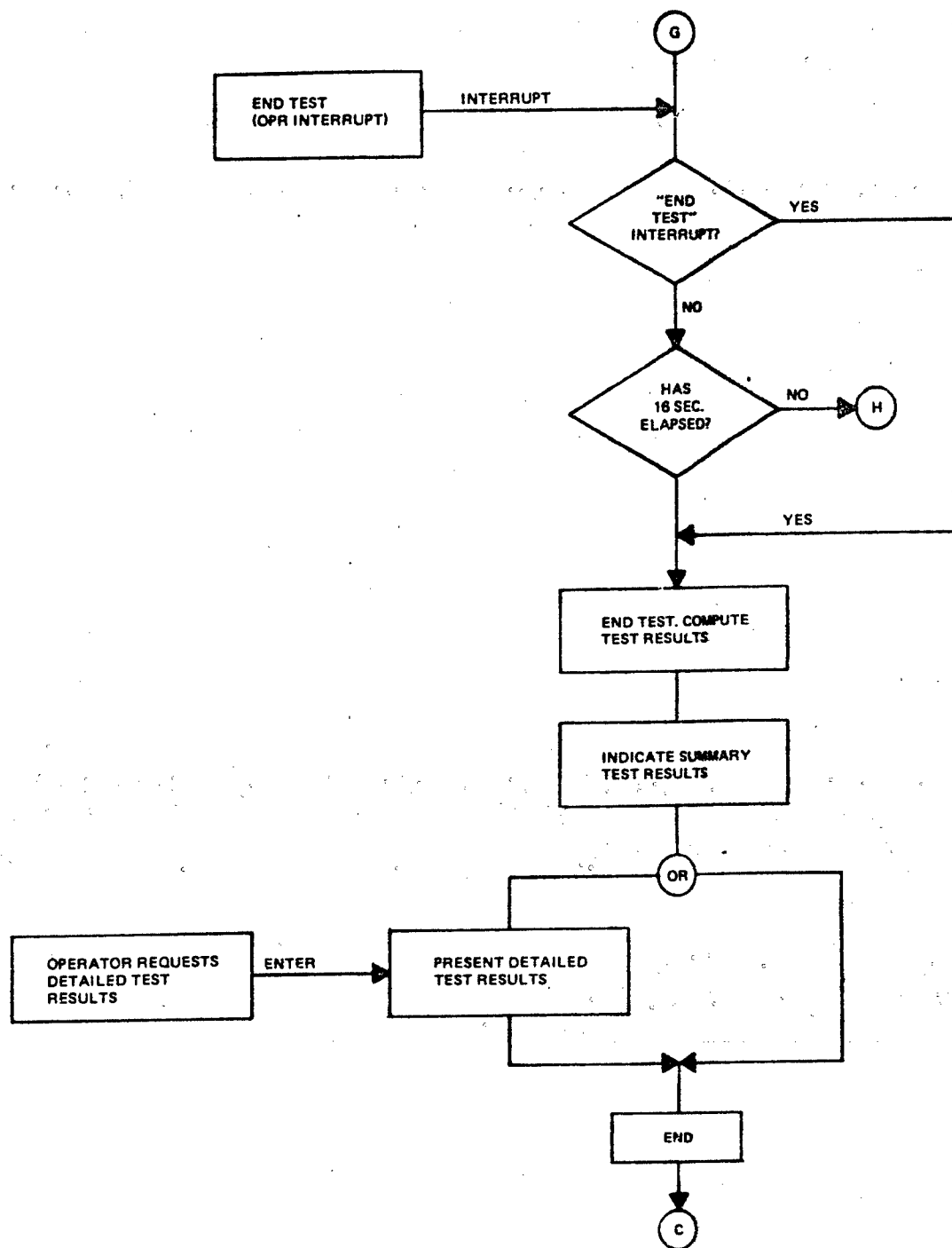


Figure 6-2. Operating Situations Flow Chart (Sheet 5 of 5).

During an engagement, using the Semi-Automatic Fire Control Mode, both initializations are necessary. Prior to target detection during the Scanning Phase, the sensors and gun are both slaved to the acquisition sensor and, hence, follow this sensor as the Gunner searches for the target. Servo outputs from acquisition sensor go directly to the tracking sensor head so the sensors can follow the search; these signals also go to the gun servos which similarly follow the acquisition system as it moves.

Following initial target detection by the FLIR, the target Acquisition Phase is entered. It is assumed that the laser does not radiate until after a target is detected in order to keep each AFAADS unit passive as long as possible. Enemy detection of the unit is thus reduced. As the FLIR obtains a good signal response, it automatically reduces its field-of-view (FOV) to the narrow value used in tracking, compared to the wide-open search value. The FLIR angle tracking circuits take over, releasing the sensor head from its slave to the acquisition system. Angular tracking errors are rapidly reduced. Target detection by the FLIR also signals activation of the laser. As the angular errors are reduced, the target comes within the field-of-view (FOV) of the laser and is detected. (The FOV of the laser is almost an order of magnitude less than the FLIR, even when the latter is in the narrow tracking mode.) Following detection, the laser range gate "opening" narrows to the tracking width about the target. Three-dimensional tracking data is now available for the computer. Under the present processing concepts, one second of data is required to fill the tracking and smoothing filter.

The end of this time the system enters the Tracking Phase. Ballistic solutions are figured and gun lead angles ordered. If necessary, sensor regeneration can begin. Once the maximum effective present range of the chosen projectile type is reached, FIRE BEGIN is indicated by the computer. This condition states to the Operator that if firing is commenced, the first projectiles will reach the target at their maximum range. This assumes the target will continue to fly the predicted course during projectile flight time. The Operator then fires at will until "CEASE FIRE" lights, indicating the target is then out of range. The actual firing remains at the Operator's command; the computer merely shows the first and last time such fire would be effective (or ineffective).

During the period that computer/automatic operations are being used by the system, projectile sensing is done, assuming the projectiles are in the FOVs of the laser and FLIR, and "bias errors" are eliminated by use of special algorithms. These algorithms "close the firing loop", insuring very accurate fire.

At any time, the Operator may interrupt the computer. One reason might be that a "hit" was made; another is that a new, more threatening target is approaching (Operator presses BEGIN NEW TARGET); and, finally a "Cease-Fire" might have been indicated, signalling to the Operator his fire will no longer be effective. Operator interruption from the Semi-Automatic Fire Control Mode generally causes the system to return to the Standby II Mode. However, if "Begin New Target" action is taken, the AFAADS will re-enter the Scanning Phase of the Semi-Automatic Fire Control Mode in order to detect and engage the new target.

The Test Mode of the system is comprised of two separate phases. First, the computer does a self-test of its own hardware. It indicates these results and waits for initialization of the second phase, or Dynamic System Test. After the Gunner has completed test initialization, the actual test begins, whereby a "canned" program is used to simulate a target. The Operator shoots live ammunition and monitors the projectile fire to see how the system is working. Generally, there should be few misses; and, after observing for a time, he may interrupt the computer and wait for test results to display themselves. He will then order the system to return to Standby II Mode. If he desires a longer period of testing, an entire test may be run (time limited), and eventually the computer and system will stop itself automatically and display results of the complete test.

### 6.3 BASIC PARAMETERS FOR THE AFAADS GUN SYSTEM

To provide a definitive and tractable basis for the digital computer sizing task, specific AFAADS system parameters were selected. These are listed below. They may not represent an optimum set: but it is believed that modifications to the listed values will not change the conclusions reached with respect to the digital computer in any major way. Different processing algorithms might change the memory requirements somewhat; but the chosen computer design uses a highly modular system, easily capable of expansion or contraction.

The AFAADS parameters listed below cover each of the basic equipment elements with which the computer interfaces.

#### 6.3.1 Gun and Gun Mount

The chosen gun and gun mount is based upon the 35 mm Oerlikon gun, primarily because data was available and its general characteristics fit the AFAADS anti-aircraft mission requirements.

- A. Gun:
  - 1) Type: Oerlikon 35mm
  - 2) Caliber: 35mm
  - 3) No. of Barrels: 2
- B. Ammunition:
  - 1) Armor-piercing high explosive
  - 2) Armor-piercing tracer
  - 3) High explosive anti-tank
  - 4) Test
  - 5) Subcaliber, high velocity
- C. Fuze: Contact
- D. Projectile Velocity:
  - 1) Initial
    - i. Standard caliber: 1175 m/sec
    - ii. Subcaliber: 1500 m/sec
  - 2) Final (at maximum effective range): 330 m/sec
- E. Range and Flight Time (Standard caliber)
  - 1) Effective (defined when remaining velocity is 110% of speed of sound): 4.5 km in 7.35 sec (Note 10 sec maximum is used in computer sizing).
  - 2) Maximum: Governed by self-destruct at  $11 \pm 2$  sec (Ref. 5)

- F. Dispersion: 1) Angular: Assumed to be 3 mr, circular, normally distributed with no appreciable change with range. May want to add artificial dispersion later. This dispersion is used in the Kalman filter algorithms for projectile miss bias correction.
- 2) Velocity: 0.5% of muzzle velocity based on analysis in Volume I and Reference 3. For AFAADS gun with:
- a. Standard Round: 6 m/sec
  - b. Subcaliber Round: 7.5 m/sec
- G. Gun Mount Vibration: Assumed small; no data. Rate gyro stabilization of the sensor provides isolation for target data. Computer will not be isolated.
- H. Alignment Accuracy: Assumed good, no data.
- I. Maximum Slew Rates: (Parenthetic values are for a tank with two guns and a doppler radar)
- 1) During target acquisition phase:
    - i. Azimuth: 1600 mils/sec (1600 mils/sec)
    - ii. Elevation: 750 mils/sec (750 mils/sec)
  - 2) During fire control:
    - i. Azimuth: 1600 mils/sec equals 90 deg/sec (1000 mils/sec equals 56 deg/sec)
    - ii. Elevation: 440 mils/sec equals 25 deg/sec (750 mils/sec equals 42 deg/sec)
- J. Aiming Accuracy (relative to sensors):
- 1) Azimuth: 0.25 mil
  - 2) Elevation: 0.25 mil

K. Ballistic Equation:

Assume an equation of the form:

$$t(p) = aD(p) + bD(p)^2 + cD(p)^3 + dD(p)^4 + eD(p)^5$$

is adequate to specify accuracies in time of flight to 0.005 second where:

$t(p)$  is projectile predicted time of flight

$D(p)$  is predicted target slant range for time of flight  
 $t(p)$

a, b, c, d, e are constants dependent on type of projectile, initial meteorological conditions, and initial muzzle velocity setting. This equation is based upon the ballistic tables for the 35mm Oerlikon gun. Additional analysis is required in this area.

L. Servo Drive Signals (Output from computer to gun servos):

- 1) Type: i. Azimuth: Lead angle relative to sensor.  
ii. Elevation: Lead angle relative to sensor.
- 2) Magnitude: i. Azimuth: -90 deg to + 90 deg or 13 bits plus sign.  
ii. Elevation: -30 deg to +70 deg or 13 bits plus sign.
- 3) Input Frequency: 10 times per second (additional analysis required).
- 4) Signal Type: Digital.

M. Servo Characteristics:

- 1) Lag: Small. Assume rapid response with no overshoot and not exceeding gun velocity and acceleration limits. Crucial parameter for closed loop fire control.
- 2) Feedback: No servo error signals into computer.
- 3) Limit Switches: Mechanical mount travel limit switches in addition to computer software limits.

N. Mount Travel Limits (Motion is relative to the tracking sensor head in both azimuth and elevation).

- 1) Azimuth: -90 deg to + 90 deg
- 2) Elevation: -30 deg to +70 deg

O. Rate of Fire: (Two guns are used in the Oerlikon system to reach the optimal rate of fire for anti-aircraft use (Ref. 5). System varies rate of fire according to range and/or motion of target.)

- 1) Maximum: 1100 rounds/minute
- 2) Length of Burst: None given. Controlled by Gunner. Details require analysis.
- 3) Burst Size: Not specified, needs analysis.

P. Control Modes

- 1) Target Search and Acquisition: Slaved to tracking sensor, which, in turn, is slaved to the acquisition sensor.
- 2) Semi-Automatic Fire Control: Lead angles relative to sensor as ordered by digital computer.
- 3) Dynamic System Test: Lead angles relative to sensor as ordered by digital computer. Sensor is directed toward simulated target.
- 4) Manual Mode: Gun slaved to an operator-controlled position. Details not determined.

#### 6.3.2 Angle Tracking Sensor

The azimuth and elevation angle position data on the target will be obtained from a Forward Looking Infrared (FLIR) sensor. This same sensor will also provide projectile miss data in azimuth and elevation.

- A. Sensor:
- 1) Type: Forward Looking Infrared (FLIR)
  - 2) Processing: Imaging device
  - 3) Radiation: None, passive sensor

B. Output Signals (to computer)

- 1) Target:
  - i. Azimuth: 0 deg to 360 deg, measured clockwise from front of vehicle.
  - ii. Elevation: -5 deg to +85 deg, positive up.
- 2) Projectile Miss (relative to target)
  - i. Interpretation: Average position of all projectiles in field of view.
  - ii. Transverse (plane through line-of-sight to target and perpendicular to the elevation angle): -4 deg to +4 deg, clockwise miss is positive.
  - iii. Elevation: -2 deg to +2 deg, up miss is positive.
  - iv. Activation: Automatic detection and processing of miss data.
- 3) Signal: Digital
- 4) Units: BAM's
- 5) Frequency: Sampled 10 times per second.

C. Response:

- 1) Target: Focuses on "hot spots"
- 2) Wander: Assumed small
- 3) Scan Rate: 30 Hz
- 4) Signal Processing:
  - i. Smoothing: None. All smoothing is a characteristic of the servo, not the sensor itself.
  - ii. Conversion: Analog to digital output buffer.

D. Field of View (FOV)

- 1) Search Phase: 20 deg x 40 deg (elevation x azimuth)
- 2) Tracking Phase: 2 deg x 4 deg
- 3) Acquisition Phase: Automatic zoom lens narrows field after initial detection.

- 4) Probability of Detection:
  - i. Inside FOV: Unity if within range. (Probably unrealistic.)
  - ii. Outside FOV: Zero, i.e., assume sharp cut-off to FOV.
- E. Range: 1) Minimum: 100 m (maximum acceptable)  
2) Maximum: 10,000 m (minimum acceptable)
- F. Accuracy of Output Data (both target and projectile miss)
  - 1) Azimuth or Transverse: 0.25 mil, 15 bit word required.
  - 2) Elevation: 0.25 mil, 13 bits plus sign in data word.
  - 3) Resolution of Projectile: No data within 0.14 mil of target.
- G. Boresighting: Laser boresighted with FLIR line-of-sight to 3 mil accuracy.
- H. Signal-to-Noise Ratio: Large (assumed), even for projectile. Needs further investigation, see also Volume I.
- I. Control Modes:
  - 1) Target Search and Initial Detection: Slaved to target acquisition sensor.
  - 2) Acquisition: With adequate target signal, FLIR generates error signals for servos, thereby disengaging lock with acquisition sensor. Laser activation signal generated.
  - 3) Tracking: Provides computer with target position signals and projectile miss data. Generates error signals for servos, which may be augmented by computer-produced regenerative signals.
  - 4) Manual Mode: Slaved to Operator control. Details not worked out.
  - 5) Dynamic System Test: Provides projectile position data to the computer. Servos driven by computer regenerative signals only.

### 6.3.3 Range Tracking Sensor

A laser will be used to obtain target range information and also projectile miss data in range. This sensor will remain on standby until the FLIR acquires the target in order to reduce the probability of AFAADS being detected. Also, the very narrow field-of-view of this sensor almost precludes target detection prior to active angle tracking.

#### A. Sensor: 1) Type: Laser

- 2) Activation: By FLIR when target is detected and angle tracking is functioning.
- 3) Radiation: Yes, active sensor.

#### B. Output Signals (to computer)

##### 1) Target Slant Range:

- i. Minimum: 100 m (maximum acceptable)
- ii. Maximum: 10,000 (minimum acceptable)

##### 2) Projectile Miss:

- i. Magnitude: -600 m to +600 m relative to target
- ii. Activation: Automatic detection and processing of miss data.

iii. Interpretation: Average miss distance of all projectiles within range gate.

##### 3) Type: Digital

##### 4) Frequency: Sampled 10 times per second, read 70 usec after FLIR data.

#### C. Response:

##### 1) Wander: Assumed small

##### 2) Scan Rate: 30 Hz

##### 3) Signal Conversion:

- i. Smoothing: None (Some lasers have internal one second smoothing in slant range and range rate. It is assumed these features are not needed for AFAADS.)

D. Tracking Gate:

- 1) Target Search: Wide open over full 100 m to 10,000 m range, i.e., 66 usec range. First return detected is assumed to be target.
- 2) Tracking: 2 usec or 600 m centered on the assumed target position. Laser tracking servo centers the gate.
- 3) Acquisition: Laser servo narrows gate from 66 usec to 2 usec about target.

E. Probability of Tracking:

- 1) Field of View (FOV): 6 mr
- 2) Detection Probability (both target and projectiles): Unity if in field of view and tracking gate, zero if outside of either (sharp cut-offs).
- 3) Signal-to-Noise Ratio: Assumed large for both target and projectiles. Needs further investigation.

F. Accuracy:

- 1) Target: 0.25 m, 16 bit word required at maximum range.
- 2) Projectile Miss: 0.25 m, 12 bit plus sign bit word.

G. Control Modes:

- 1) Target Search: In standby until FLIR detects target..  
When activated, first return in wide open gate is assumed to be target.
- 2) Acquisition: Laser range servo narrows target gate around return.
- 3) Tracking: Provides both target range and the average projectile range miss to the computer. Generates error signals for its own range servo.
- 4) Manual Mode: Provides Operator with target data. Details not worked out.
- 5) Dynamic System Test: Provides projectile range data to the computer. Tracking range gate driven by the computer about the simulated target's range.

#### 6.3.4 Sensor Mount Servos

The FLIR and laser sensors are mounted on a single rate gyro stabilized mount. The mount contains a closed loop servo system operating off the FLIR angular error signals. It can also accept computer developed regenerated angular target rates. The mount travels with respect to the AFAADS vehicle. The FLIR and laser are carefully boresighted with respect to one another.

##### A. Stabilization:

- 1) Type: Transverse and elevation rates.
- 2) Drive Signals: Transverse and elevation rates.
- 3) Sensing Equipment: Two rate gyros.
- 4) Purpose: To isolate tracking sensor from gun vibrations

##### B. Output Signals (to computer):

- 1) Magnitude:
  - i. Transverse: -90 deg/sec to +90 deg/sec, positive clockwise.
  - ii. Elevation: -25 deg/sec to + 25 deg/sec.
- 2) Signal: Digital
- 3) Frequency: Sampled 10 times per second.

##### C. Servo:

- 1) Lag: Small, assume that no regeneration is required, except for missed data points and that the slew limits are not exceeded.
- 2) Backlash: Assumed negligible.
- 3) Maximum Slew Rates:
  - i. Transverse: 90 deg/sec
  - ii. Elevation: 25 deg/sec
- 4) Maximum Acceleration Rates:
  - i. Transverse: 90 deg/sec<sup>2</sup>
  - ii. Elevation: 70 deg/sec<sup>2</sup>

D. Control Modes:

- 1) Target Search and Initial Detection: Drive signals from target acquisition sensor.
- 2) Acquisition: Automatic switch to FLIR angular error signals when available.
- 3) Tracking: Drive by FLIR angular error signals augmented by computer regenerated rate signals when target is behind an obstacle or fades. Rate gyros send tracking rates to the computer.
- 4) Manual Mode: Sensor servos slaved to Operator control. Details not worked out.
- 5) Dynamic System Test: Sensor head driven by computer to follow simulated target track.

6.3.5 Vehicle Orientation Sensors

Simple sensors are required to determine the pitch and cant of the vehicle relative to a horizontal surface. These angles are used in making the various ballistic corrections due to gravity and wind.

A. Output Signals to Computer:

- 1) Magnitude:
  - i. Pitch: -35 deg to + 35 deg, positive for front up.
  - ii. Cant: -35 deg to +35 deg, positive for right side down.
- 2) Interpretation: Vehicle first pitches, then cants.
- 3) Type: Digital
- 4) Frequency: 10 times per second.

B. Accuracy:

- i. Pitch: 0.6 deg, 6 bits plus sign bit
- ii. Cant: 0.6 deg, 6 bits plus sign bit.

C. Control Modes: Only used during fire control computations portion of the Semi-Automatic Fire Control Mode and Dynamic System Test.

#### 6.3.6 Digital Computer

The basic parameters that are assumed for the AFAADS digital computer deal with processing requirements rather than hardware. It is these processing requirements that dictate the hardware concepts and overall size.

##### A. Fire Control Processing

- 1) Target Tracking
- 2) Track Prediction:
  - i. Linear
  - ii. Constant energy in a diving attack.
  - iii. Constant turn (acceleration) attack.
- 3) Ballistic Solution
- 4) Ballistic Corrections:
  - i. Gravity
  - ii. Air temperature
  - iii. Air pressure
  - iv. Wind speed and direction
  - v. Muzzle velocity as a function of projectile type and tube wear.
- 5) Sensor Regenerative Tracking
- 6) Closed Loop Tracking:
  - i. Projectile miss detection and processing
  - ii. Bias corrections in azimuth, elevation and muzzle velocity.

##### B. Operator-Entered Parameters:

- 1) Projectile Type: One of five, see Section 6.3.1.
- 2) Muzzle Velocity:
  - i. Preset, Standard Projectile: 1175 m/sec
  - ii. Preset, Subcaliber: 1500 m/sec
  - iii. User-Entered Range: 930 m/sec to 1570 m/sec
- 3) Air Temperature:
  - i. Preset: 70 deg F.
  - ii. User Range: -20 deg to + 120 deg F.

4) Air Pressure:

- i. Preset: 1013 mb
- ii. User Range: 500 to 1040 mb

5) Wind Speed:

- i. Preset: 0 m/sec
- ii. User Range: 0 m/sec to 30 m/sec (66 mph)

6) Wind Direction:

- i. Preset: 0 deg (or indeterminate)
- ii. User Range: 0 to 360 deg.

7) Dive Angle Threshold:

- i. Preset: 10 deg., absolute value
- ii. User Range: 2 to 15 deg. , absolute value

8) Turn Acceleration Threshold:

- i. Preset: 0.3 g , absolute value
- ii. User: 0.2 g to 1.0 g , absolute value

C. Operating Modes: See end of Section 6.1.

D. Peripheral Devices:

1) Sensor Head:

- i. FLIR
- ii. Laser
- iii. 2 Rate Gyros
- iv. Servos

2) Operator's Display and Controls

3) Gun Mount

4) Vehicle Orientation Sensors

## SECTION 7

### OPERATOR CONTROLS AND SCENARIOS

Any weapon system requires human control. A good man-machine interface design greatly enhances a weapon's effectiveness, particularly under the stress of battle. Realizing this, considerable attention was directed toward providing a realistic and useful interface between the Gunner/Operator and the AFAADS digital computer. Additional effort is still required, particularly to add other displays and controls such as the fire trigger and manual fire controls.

The results of the man-machine concepts analysis are presented in two parts:

- a. The Operator's Panel from a hardware and functional viewpoint.
- b. The employment of the panel, as demonstrated by typical scenarios.

#### 7.1 THE OPERATOR'S DISPLAY PANEL

The computer display panel for the AFAADS Gunner/Operator is illustrated in Figure 7-1. As will be discussed in Section 10 on the computer hardware, this panel will form the front surface of the digital computer; thereby reducing cabling and connector requirements. The layout of the displays and controls is directed toward simplifying its use, its maintenance, and spare parts.

With respect to maintenance and spares, only four different components are used:

1. Pushbuttons
2. Indicators
3. Numeric light emitting diode (LED) displays
4. Rotary potentiometer (one only).

The same pushbutton would be used, whether an indicator is required or not. In the latter case, the indicator would remain disconnected.

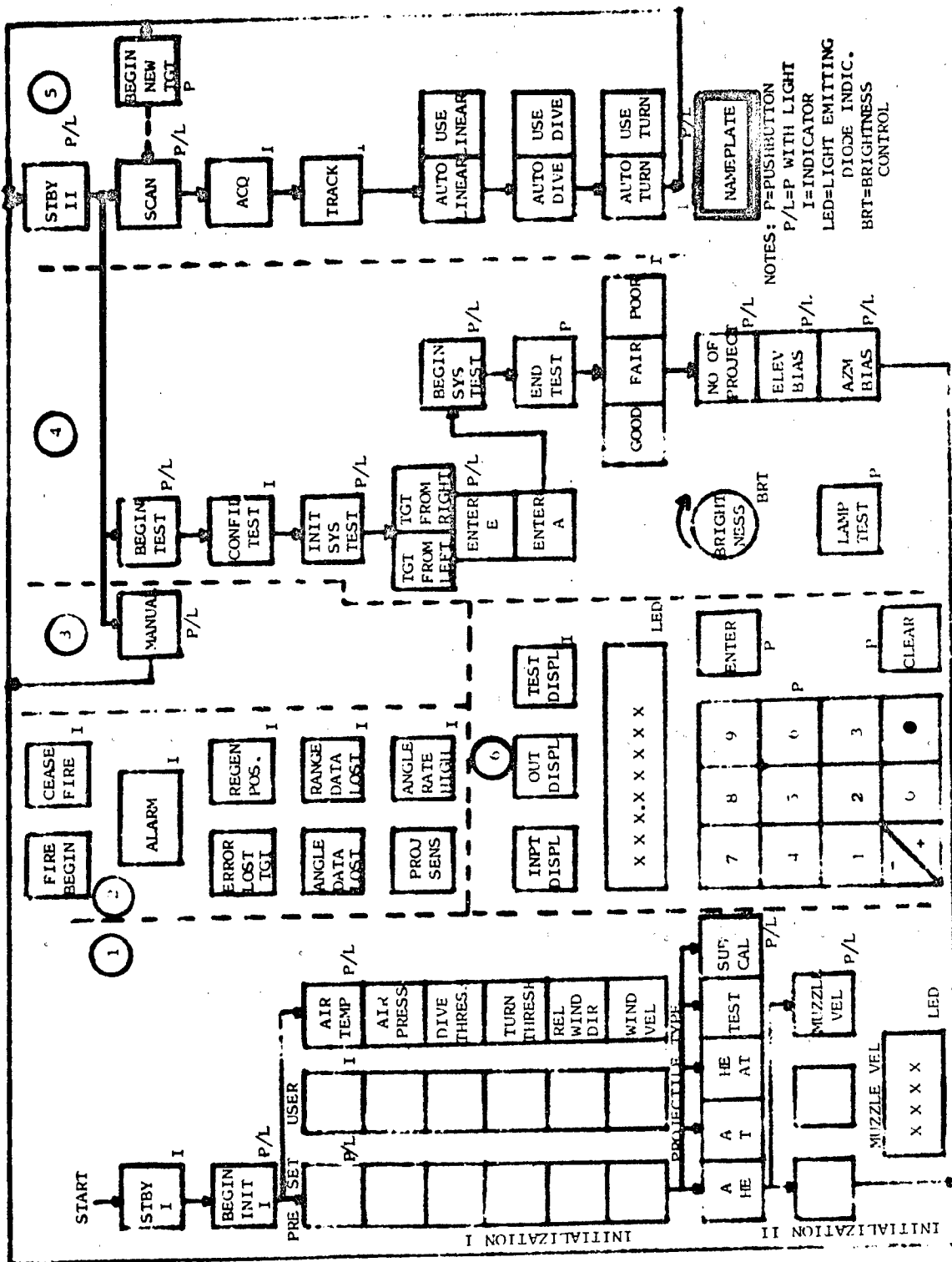


Figure 7-1. Operator's Panel

Employment of the Operator's Panel is facilitated by several techniques:

1. Displays and controls are grouped into functional areas.
2. Operator use is enhanced by lines and arrows from one operation to the next.
3. The flow is generally from top to bottom, right to left, with certain critical displays in the top center.
4. There is minimum interference between operator activation of controls and operator observation of the keyboard.

The six functional areas on the keyboard are separated by dotted lines on Figure 7-1 and keyed by numbers in small circles. These areas correspond more or less to the various operational modes of AFAADS. The general functions associated with each area, by number, are:

1. Initialization. The displays and controls required to enter the Initialization I and II Mode data.
2. Alarms and Fire Control Status. A group of nine displays closely associated with the status of the Semi-Automatic Fire Control Mode during gun fire. The computer inoperative alarm is also in the group.
3. Manual Fire Control. The single pushbutton/indicator that places the AFAADS into manual fire control. Since the computer is placed outside the fire control loop, no other displays are required on the computer panel. Other controls and displays are required, however.
4. Test. The displays and controls associated with the Test Mode, both the operator-initiated confidence self-test of the computer and the overall dynamic system test.
5. Semi-Automatic Fire Control. The displays and controls required for the Semi-Automatic Fire Control Mode operation, as opposed to the status displays of Area 2 above. This also includes the Standby II Mode control and indicator. Most AFAADS operational missions will utilize this area.
6. Keyboard. The group of keys and readouts required for entry and display of numeric data. These are associated with the initialization and test functions of the two areas on either side (Areas 1 and 4).

Overall, when AFAADS is first activated at a new defense site, the operator deals with the controls on the left portion of the panel to initialize the system. He then places the gun system in Standby (top right). Most fire missions then involve the right-most area on the panel. Status of equipment operation during the fire mission is provided by the indicators, top center (Area 2).

Use of this panel is now illustrated in a typical scenario, starting with system initialization (Subsection 7.2) and passing through standby to actual gunfire (Subsection 7.3). Scenarios covering the other modes of manual and test are then briefly discussed (Subsection 7.4).

## 7.2 INITIALIZATION SCENARIO

Initialization uses the indicators and pushbuttons in Area 1 of the Operator's Panel (Figure 7-1). There are two types of initializations which take place.

### 7.2.1 Beginning Initialization

The AFAADS System is to be mobile and to be usually used near a point to be defended. The apparatus could be towed or mounted on a tracked vehicle. Once the system is in place, the computer and gun system must be warmed up for use. Although radar systems may take a period of time before they are ready for use, a long warm-up time may not be necessary for the FLIR/laser sensors proposed here. The computer is ready for use immediately after it is turned on. Assuming it has been turned on, we proceed to the beginning of initialization.

Once turned on, "STBY I" or standby-without-data will be indicated. The Operator would press "LAMP TEST" to light all indicators, showing all lights to be in order. He could also adjust "BRIGHTNESS" dial if he desired, but would probably not do so until he has finished with Initialization (I and II).

After the Lamp Test and replacement of defective bulbs (or Light Emitting Diodes - LED's), he would proceed to Initialization I.

### 7.2.2 Initialization I and II

The Operator would now press "BEGIN INIT I", readying the computer for input data. This will release from software interlock the various initialization parameter pushbuttons. It will also show the Operator how these parameters were designated during the last fire mission by lighting the appropriate Preset or User indicators and the previously selected projectile type switch. (see Figure 7-1, Area 1).

If the system is needed to be used immediately, the Operator could press STBY II, followed by SCAN, and begin to use gun. Hopefully, he will have enough time to set some values fitting the situation more closely. Specifically, the time is probably in the morning; the previously set values are probably not quite accurate. Taking into account a minimum warm-up time of fifteen minutes and these conditions, he would probably change some of the values above.

For the following discussion, we shall assume that the previous values were all preset ones. Then, following the arrows on Figure 7-1:

1) The air temperature is assumed to be closer to 50°F. than the 70°F. preset value. To change the air temperature in the computer, he would press AIR TEMP. "70" would light on the keyboard display to his right. Also, OUT DISPL would light, indicating output data was indicated on the display.

To enter a new value, he would first press CLEAR. The LED display would go blank and OUT DISPL indicator off. Then, he would press "5", the first digit of the temperature he desired. The LED display would show a "5" in the right-most position. Next, he would press "0". Satisfied that 50 degrees was the new AIR TEMP, he'd press ENTER. The INPT DISPL indicator and the USER light left of AIR TEMP would light, indicating the value has been entered.

2) To verify the preset air pressure is the desired value, he would depress AIR PRESS. After pushing AIR PRESS, the AIR TEMP, 50, and INPT DISPL indicators would all go out. "1013" would light on the display as well as OUT DISPL. Satisfied that "1013" was the correct air pressure in milibars, he would proceed to the next entry.

3) The next value to be altered in this hypothetical initialization would be the dive threshold. It is not known how the sensor system

would degrade. However, let us assume such a degradation. The resultant noisier data from the sensors will probably cause more irregularities in target prediction than is desirable. Therefore, the Gunner should have the capability to reduce the effect of this noise on target prediction. He does this by changing the dive and turn prediction thresholds, making the computer reject the noise before prediction is made. The case here is to modify this threshold from a 10 degree angle to a 12 degree angle, decreasing the probability of "jumping" and inaccurate firing.

This threshold could be misinterpreted. Without getting too deep into the logic of the computer, the use of a "threshold" ought to be explained. Basically, the computer will monitor whether or not to use a "constant energy" term for prediction during a dive. If the dive angle of the target track goes over 12 degrees, the computer would switch to the dive prediction method. If the plane is in a slow (non-accelerating) dive, the computer will predict a straight line path; if the rates show the target is diving, however, a term reflecting a "dive" will be inserted into the "predicted-position equation".

Following a procedure similar to that applied to changing air temperature, the Operator would:

- a) Press DIVE THRESH (Dive Threshold).  
DIVE THRESH, PRESET, and "10" would be indicated. Also, OUT DISPL would be indicated.
- b) Press CLEAR.  
OUT DISPL and 10 would be extinguished.
- c) Press "1".  
"1" would light in the LED display.
- d) Press "2".  
The "1" would shift left once; "12" would then be displayed.
- e) Satisfied 12 degrees was new dive angle threshold, he'd press ENTER.  
PRESET would go out. USER, INPT DISPL and "12" would light, showing a nonstandard user entry of 12 was entered.

4) Turn threshold would probably be changed for the same reason as dive angle was. The standard value of 0.3 g's would be changed to 0.4 g's.

5) We'll assume no wind; therefore, wind direction would remain 0 degrees.

6) Wind velocity would be zero. Thus, the effects of wind will be ignored in the ballistics solution.

7) The projectile type would probably be the armor-piercing high explosive (A HE) round (unless the gunner ordered a test).

8) Because the barrel of this weapon is assumed to be worn, we expect the muzzle velocity to be lower than usual. The four digits under the MUZZLE VEL indicator read "1180" for 1180 m/sec. muzzle velocity. The PRESET indicator to the left of the MUZZLE VEL pushbutton is lit, signifying 1180 m/sec. muzzle velocity is normal for the armor-piercing, high explosive round. The Gunner will reduce this by 20 m/sec., to cause computer to account for barrel wear. He would then enter 1160 using the method previously described, save for one difference. The Gunner will make an error in entering the new value, to illustrate the system reaction to an erroneous entry.

a) He presses MUZZLE VEL.

MUZZLE VEL, PRESET, 1180, and OUT DISPL are all indicated. (1180 has been duplicated from four digit display onto eight digit display.)

b) He presses CLEAR.

The OUT DISPL and 1180 go out on the keyboard. The number remains on the four digit display.

c) He presses 2 (by accident).

The eight digit display shows "2". (The "1180" on the four digit display does not change.)

d) He presses "1", "6", and "0", one-by-one, each time the digits automatically shift left.

e) "2160" is now displayed. He presses ENTER.

f) Nothing changes.

g) The error is obvious. "2160" is not an acceptable value for this projectile type; the number will not "go in".

8) (continued)

h) He presses CLEAR.

Display goes blank.

i) He presses "1".

"1" is shown on eight digit display.

j) He presses "1", "6", and "0" in that order; each time, the digits shift. "1160" is now shown on the eight digit display, while "1180" is shown on the four digit display.

k) He presses ENTER. INPT DISPL and USER light.

PRESET goes dark. Both the four and eight digit displays show "1160". This signifies that the new, lower muzzle velocity will be used in solving the ballistics problem.

He now may adjust the brightness of the display if he wishes to do so. The arrows on the display indicate the next stage is Standby II, standby-with-data. He presses STBY II, and system goes to a standby status, ready for a target. It is now fully initialized for an engagement.

At the press of STBY II, BEGIN INIT I has gone out. As initialized, these conditions and indications are present:

- 1) Air temperature is nonstandard. USER to the left of AIR TEMP is lit.
- 2) Air pressure is standard. PRESET to the left of AIR PRESS is lit.
- 3) Dive threshold is nonstandard. USER to the left of DIVE THRESH is lit.
- 4) Turn threshold is nonstandard. USER to the left of TURN THRESH is lit.
- 5) Relative wind direction is standard (Because the wind velocity is zero, the standard value makes no difference). PRESET to the left of REL WIND DIR is lit.
- 6) Wind velocity is standard of zero meters per second. PRESET to left of WIND VEL is lit.
- 7) Projectile type is armor-piercing, high explosive. A HE is lit.
- 8) Muzzle velocity is nonstandard. USER to left of MUZZLE VEL is lit.

- 9) Four digit display below MUZZLE VEL is lit, indicating user-entered value of "1160".
- 10) Eight digit display and keyboard are not lit, and software locked-out.
- 11) Standby-with-data is indicated. STBY II is lit.
- 12) No other indicators are lit.

### 7.3 FIRE MISSION SCENARIO

The main mission of AFAADS is to provide anti-aircraft fire when needed for forward areas, especially to give a point-defense capability. This system, during a fire mission, will give the Gunner the ability to engage different types of targets and to alter the AFAADS system to match the engagement by the right orders to the computer.

Two examples are built into the current computer concepts. In the first, the gunner/operator can modify the target prediction algorithm to suit the observed target track. Linear, constant dive, and constant turn predictions are available. These are detailed later in this scenario. The second is a computer capability which, hopefully, can also be realized in the gun itself, namely to change ammunition type during a fire mission. At long range a high velocity sub-caliber round is more appropriate than at, say 3000m range, full caliber ammunition would be employed, with its greater lethality.

As presently planned, AFAADS provides defense against three different aircraft missions. The basic defensive mission is to fire on a single aircraft making a direct attack on a target, say, about 200 meters from the gun. In the second attack situation, one may expect two or more aircraft attacking at once, one strafing the defended point carrying no external weapons, and a second aircraft following up to 15 km away (or about a minute "behind" the first) carrying iron bombs for release upon the defended point. Lastly, some hostile aircraft may inadvertently pass through the range envelope of the AFAADS gun by flying below ordinary radar coverage, and hence in gun range.

Sections 7.3.1, 7.3.2, and 7.3.3 describe the system scenario for a single attacking aircraft. Section 7.3.4 details system use during a

multiple aircraft attack. The last section, Section 7.3.5, describes use of AFAADS for a passing target.

The reader is referred to Figure 5-2, "Typical AFAADS Fire Mission Time Event Plot". The events on that figure are used to develop the first three sections of the Fire Mission Scenario.

#### 7.3.1 Acquisition and Lock-On

Section 7.2 developed the initialization of the AFAADS System. At the end of that phase, the gun system was in Standby II, a standby with proper initialization data. Referring to the time scheme on Figure 5-2, the system is still in standby. At  $t=40$  seconds, the Gunner receives a radio report: Single hostile aircraft, probably going to attack the point target defended by the AFAADS gun, is flying due west and at very low altitude. (Time,  $t$ , is measured from the point of closest approach.) At this time, the aircraft is 12,000 m east of the defended point and approaching. Then, at  $t=0$ , the aircraft is 283 m from the gun and is as close to gun as it will be; at  $t=15$ , the aircraft is leaving the area and is 4500 m away. The aircraft in this fire mission is traveling at 300 m per second (520 knots and near Mach 1) and at 200 m constant altitude.

Because the computer is already initialized, the Gunner would check the parameters, but probably would not change any setting. The Initialization I settings, as set in the Initialization Scenario of Subsection 7.2, show non-standard AIR TEMPERATURE ("USER" to left of "AIR TEMPERATURE" is lit), standard or preset AIR PRESSURE, non-standard DIVE THRESHOLD, non-standard TURN THRESHOLD, and preset wind direction and velocity; i.e., no wind. These Initialization I settings are differentiated from Initialization II settings by their slow variation. The projectile type and muzzle velocity may be changed during and as a part of a firing mode.

The Gunner, since he has not used the gun or changed ammunition, is assumed not to change the presently indicated Initialization II conditions: projectile type has been set at "A HE", armor-piercing, high explosive; and muzzle velocity as 1160 m/sec, which is a "USER" entry. Thus, he only checks the indicated projectile type and muzzle velocity. Satisfied the

computer is ready to use, he presses "SCAN" and begins to search for the aircraft.

Using his acquisition system, the Gunner searches the eastern horizon, realizing the aircraft is no more than two or three degrees above the horizon. The gun and FLIR sensor follow the acquisition system. The laser is inactive and in standby. The FLIR, being a passive sensor, decreases likelihood of detection and of countermeasures being taken until the target has been acquired, particularly if a passive acquisition system is also employed.

The Gunner detects the target; the FLIR detects the target as well. Angle tracking commences. The ACQUISITION indicator lights, thereby showing the target is being followed. The FLIR hardware is assumed to recognize and signal this condition. The FLIR zooms from a  $20^{\circ} \times 40^{\circ}$  FOV to a  $2^{\circ} \times 4^{\circ}$  FOV. The laser is activated, and the target is detected by this sensor as well.

In this scenario, when the aircraft is 8500 m away, or at  $t=28$  seconds, the system goes automatic, laser and FLIR have successfully locked on target. The gun remains slaved to the sensors until a fire control solution is generated.

Once the laser has locked onto the target, the computer begins to accept data from the sensors. Prior to this, it has "refused" to do so, since the data was not complete; i.e., no range was available until the laser successfully found the target. After 11 complete data points (one second of data) are received, the computer starts to develop the proper gun lead angles. The gun mount now will start to follow these orders rather than be slaved to the sensor head. The Operator's Panel TRACK indicator lights, indicating target tracking and a fire control solution.

Once "TRACK" lights, all the indicators associated with an engagement may light. These will be discussed in the remaining sections. The prediction mode indicators and pushbuttons would be used first. Prediction refers to digital machine computing and firing on a predicted position of the aircraft. Given the aircraft's position and rates, the computer predicts where the aircraft will fly next. Then, using these

values, it solves the ballistics-prediction and aircraft-prediction problem iteratively, and eventually places the gun at the proper angles to have the projectiles hit this predicted position. If both predictions are good, the projectile will hit the aircraft. Prediction in the discussion refers to aircraft-position prediction.

As stated in the Initialization Scenario, Section 7.2.2, the sensors are assumed to be noisy. Knowing this, the Gunner may order his own prediction method. Therefore, the Gunner may override the automatic prediction modes. He does this by pressing "USE DIVE" and "USE TURN". Depending on the sort of data the sensors received during the 1.0 second period of preliminary sensing, "AUTO LINEAR", or "AUTO DIVE" and/or "AUTO TURN" will have been indicated. These may be overridden. If so, a new fire control solution, using the target coordinates and rates and the orders of the Gunner, will be computed. Once computed, the gun will be ordered to the new direction. From  $t=27$  to  $t=22$  (see Figure 5-2 again), the gun will maintain a small lead angle (less than a degree), and the computer will compute prediction and lead angle orders. When the aircraft is about 22 seconds from the point of closest approach ( $t=22$ ), the system enters its Firing Phase.

#### 7.3.2 Firing on Target

The initial tracking phase includes a 5 second "dead" time while the target is coming into maximum effective range of the gun. This time is used to press the two prediction mode buttons, and to perform identification and verification procedures that verify that the aircraft is indeed hostile.

##### 7.3.2.1 Indications During an Engagement

Referring to the Operator's Panel, Figure 5-6, one notes numbered areas. Area 2 contains the Fire Control Status Indicators. They will be referred to occasionally during the remainder of the fire mission scenario. Examining this area from top to bottom, one finds:

- 1) FIRE BEGIN and CEASE FIRE indicators. Placed here, these indicators command the most obvious position of the Operator's Panel. They indicate "target in range" or "out of range", respectively; the most crucial indications given by an automatic system such as this. Both will be discussed further in this section.

2) ALARM indicator. This indicator is strategically placed for the same reason as the indicators above it. If the computer should find it is not operating correctly, "ALARM" will light, indicating some major failure in the system. The malfunctions detectable by the computer include:

- a) Some peripheral device does not "answer". The computer "speaks to each piece of hardware it is linked to. If one, say the rate gyro, will not "answer" the periodic "questioning" by the computer, the ALARM indicator lights, signifying a peripheral device (gyro sensor) will not "answer".
- b) Critical failure in the computer. A self-test procedure, continuously executed, has the computer check itself. If some failure is evident to it, an "ALARM" will be indicated.

3) ERROR: LOST TARGET and REGENERATING POSITION indicators.

If both FLIR and laser target data are lost simultaneously, once the target has been acquired, "REGENERATING POSITION" will light\*; the computer assumes the simultaneous loss of both data means the target is behind an obstacle. Ideally, there will be no hills, trees, etc. between the sensors and the target. One cannot hope for perfect firing areas; therefore, this indicator is necessary. The ERROR: LOST TARGET indication may have several meanings:

- a) If the computer regenerates position for two seconds, and no target is in view, it is assumed lost. The computer should indicate the system may be losing the target. If the target cannot be seen by the

\*Regenerating position. The computer, with sufficient data, may predict future target position for some period of time, probably longer than two seconds. It does this by using angular and range rates and position to predict where the target will fly in the next tenth of a second. This is meant to be used only for a short time because it is an artificial, short term "fix". This kind of prediction is on a tenth-of-a-second basis, versus prediction for firing of gun in which the computer predicts target position up to seven seconds into the future.

3) a) (continued)

Gunner using an optical sight or by the FLIR/laser sensor, it is probably still behind some obstacle. The Gunner has a choice: if he thinks the target is departing, he may press STANDBY II, and wait for a new target; or if the target is behind a minor obstruction, he may allow the gun and sensors to continue to slew, hoping the target will appear in their FOV very shortly\*.

- b) The aircraft disappears, then reappears optically; but not in the FLIR FOV, it has either made a drastic maneuver, or the sensors have some poor data, made worse by the regeneration algorithm in the computer. (In the two seconds delay before the warning light comes on, the computer has used all of its previous data twice and is doing the best with it that it can. Because it has generated position this long, any major maneuver by the aircraft while behind the obstacle, or bad information received just prior to the disappearance will cause the sensors to lose the target. This is especially true at short range.) The Gunner must reacquire the target. This is rather drastic unless he is well-trained, since the target is probably by now flying at close range with a very high angular speed.

\*A little calculation shows that if there is a hill 400 m east of the gun and the aircraft flies a straight-line path, as in our example, the target could be lost. If the hill were 60 feet high and a tree was only 10 feet wide at its base, the target would not be visible for three seconds! The Gunner must use his own judgment here.

- 4) ANGLE DATA LOST and RANGE DATA LOST indicators. The laser and FLIR data may be lost occasionally. These indicators convey this information to the Gunner. If they are continuously lit, the FLIR (angle) sensor or laser (range) sensor is malfunctioning or some of their interfaces are not working properly.
- 5) PROJECTILES SENSED and ANGLE RATE TOO HIGH indicators. Both deal with the "closing of the loop" with this AFAADS System. Since projectiles will be sensed when they pass close to the target, this ought to be indicated, as a check on the computer as well as the sensors. Of course, the main reason for "closing the loop" is to insure a tight shot pattern about the aircraft, very accurate firing, and an eventual hit. As the aircraft gets close to the gun, the more data becomes inaccurate, due to the high slew rate of the sensors. These indicators and reasons for use are discussed in more detail later.

#### 7.3.2.2 Prediction During an Engagement

Returning to our scenario, the Gunner has entered threshold values for predicting target position during the initialization activity. Intuitively, this operation has the effect of causing the computer to ignore the bad data it might be receiving and instead use more conservative prediction methods. In our example, the target will fly a straight line, hence, only linear prediction need be used; but the Gunner does not know this a priori. He, therefore, presses the USE DIVE and USE TURN pushbuttons. These enable the computer to use a dive or turn as a predicting equation for estimating target position. When these indicators are lit, the computer will apply the DIVE THRESHOLD and TURN THRESHOLD values entered earlier by the Operator. If the Gunner had pressed only one or the other, or none at all, the computer would react as follows:

- a) If he'd pressed only "USE DIVE" or "USE TURN", the computer will apply only dive or turn thresholds to predict the target's future position and not the other one.
- b) If he'd pressed only "USE LINEAR", only straight line prediction would be used. That is fine for our example, but

b) (continued)

usually not true. See Section 7.3.5 for details of when the "USE LINEAR" prediction could reasonably be used.

c) If he pressed none of the prediction buttons, all automatic logic would be used by the computer to predict future target position. In this case the logic would be identical to that used with both USE DIVE and USE TURN pressed. We have an over eager operator who wants to assert himself.

#### 7.3.2.3 Firing on Target

The immediately preceding discussions were to explore possible indications. The remaining discussion deals with this scenario only.

At  $t=22$  (see Figure 5-2 again), the first indication to fire is given. "FIRE BEGIN" informs Gunner he may fire. The aircraft is over 6500 m away, but by the time the projectiles reach it, the projectiles and aircraft should be at the same range of 4500 m and, hopefully, the same position.

Figure 5-2 calls this initial lag time, "Pre-Bias Correction Fire". No projectiles can be sensed at this time, since none have reached the target.

Once the aircraft is 4500 m from the gun, "PROJ SENS" should first light, informing Gunner that the sensors and fire control system are now measuring and correcting for uncompensated muzzle velocity error or bias azimuth error or bias and elevation error or bias in the gun. Approximately 130 projectiles can be observed, and their average corrections applied to increase the aiming accuracy. When  $t=11$ , projectile sensing is no longer possible. At that time, "ANGLE RATE HIGH" will indicate. Projectile miss corrections will not be applied again during this engagement due mainly to the high angular velocity and close approach of the aircraft.

By now, however, the statistics on 130 shells have been taken, and the next 19 seconds of fire should be highly accurate.

In the meantime, "FIRE BEGIN", "USE DIVE", "USE TURN", "PROJ SENS", and "ANGLE RATE HIGH" have all been indicated. Eventually, as target leaves and reaches about 2300 m from gun, fire becomes ineffective.

### 7.3.3 Cease-Firing and Disengagement

Once the target is past the point of closest approach, "t" becomes negative in this scenario, indicating the target is leaving the area of the AFAADS gun. At  $t=-8$ , range of about 2300 m, CEASE FIRE will indicate, showing that target is now out of effective range of the weapon. At  $t=-15$ , the last projectile reaches target at maximum effective range of 4500 m.

Once CEASE FIRE is lit, the Gunner would stop firing and release the system. If no other targets are in the area, he would press "STBY II," and return to a standby mode. At the conclusion of the fire mission, "USE DIVE", "USE TURN", "ANGLE RATE HIGH", and "CEASE FIRE" would all go off. Only "STBY II" and the several Initialization I and II indicators would remain lit, indicating standby data. If there were other targets to fire upon, the Gunner would press "BEGIN NEW TARGET" and proceed as in the next subsection.

### 7.3.4 New Target Sighting

The Gunner would press BEGIN NEW TARGET in either of the two cases below:

- 1) A new target, probably a bombing aircraft (versus the strafing aircraft in the previous three sections) is reported to him. In that case, he would press BEGIN NEW TARGET, and search the horizon again. "SCAN" would light automatically and the acquisition/lock-on phase just described would be reentered.

- 2) In the case of a multiple aircraft attack, a bombing aircraft might follow a few seconds after the strafing aircraft. He would lock-on and fire on the first aircraft up to about the point of closest approach. Then he would break-off this attack by pressing BEGIN NEW TARGET. He would then acquire and fire on this second target, since it is now more of a threat to the defended point than the first.

In either case, the system would act as it had previously. The bias correction data that it had processed on the previous firing would be saved, thereby insuring more accurate fire at longer range.

### 7.3.5 Passing Targets

Occasionally, an aircraft will accidentally enter the AFAADS air space. The plane is probably going to attack more deeply into friendly territory, or perhaps it has already attacked and is returning to its home

base. For whatever reason, a hostile aircraft may fly over, up to 8000 m altitude. These aircraft are within AFAADS range but must be dealt with specially.

The aircraft acting as "passing targets" will normally fly a straight line path across the AFAADS defended area, not attacking, and not being evasive until fired on. In such case, the Gunner could acquire and lock-on such aircraft. Then he would press USE LINEAR, causing the computer always to predict along a straight line. Data seems to indicate that since only minor evasive maneuvers are expected, the best counter is "a lot of lead directly in his path." Therefore, the Gunner would fire in front of him as he flew away. Of course, if the target started to return to attack, the Gunner would now have an "ordinary" target. He would press USE DIVE and USE TURN to signal that fact, or go to the fully automatic prediction method by deactivating USE LINEAR.

Except in special situations such as just described, the USE LINEAR pushbutton would not be pressed.

#### 7.4 OTHER MISSION SCENARIOS

To complete the description of the operational capabilities of the Operator's Panel on the computer (Fig. 7-1), two other special situation scenarios are required.

##### 7.4.1 Manual Mode

The manual mode scenario would be used under special conditions such as computer failure (including any of the input/output buffers) or weak noisy target data. In the manual mode, the Gunner/Operator is given complete control of the sensors, gun mount, and gun. He performs the fire control solution.

Since the digital computer is out of the fire control loop, the operational scenario relative to the Operator's Panel on the computer is very simple. The Operator simply pushes MANUAL (Area 3 on the panel layout of Fig. 7-1). This action will:

- a) Place the digital computer in the Standby II Mode, in case the operator wishes to switch to the Semi-Automatic Fire Control Mode, or another mode.

- b) Release the FLIR/laser and gun from computer control and place them in manual control of the operator.
- c) Light the MANUAL pushbutton, and extinguish the other mode buttons on the panel.
- d) Activate the manual displays and controls. What these manual displays and controls are remains to be determined. They are considered to lie outside the present effort to develop the AFAADS digital computer concepts and sizing.

#### 7.4.2 Test Mode

There remains the Test Mode Scenario. The AFAADS computer panel provides in Area 4 all the displays and controls required to operate the system in the confidence self-test and dynamic system test operations. The scenario describing the operation of these several pushbuttons and indicators is included in the test description in Section 13 of this volume.

## SECTION 8

### DATA PROCESSING PERIPHERALS

The fire control computations performed by the AFAADS digital computer in each of its modes are based upon data received from the various sensors and upon instructions from the operator. Gun orders, sensor servo drive signals, and status information are outputted as a result of the computations. Overall, these input and output signals are sent by or received by the computer peripheral devices.

This section discusses the AFAADS computer peripherals from the computer processing viewpoint. Two topics are covered:

- a. Peripheral Devices, a general description of each computer peripheral device (Subsection 8.1).
- b. Specific Interface Signals, a detailed listing of the signals involved in each interface and the computer interface word (Subsection 8.2).

The maximum word size involved in this input/output processing is one primary determinate in the choice of a 16-bit word for the computer.

#### 8.1 PERIPHERAL DEVICES

The AFAADS digital computer interfaces with four different equipments. These have, in turn, eight different computer interface peripheral devices (Table 8-1). The eight devices are used by the computer (a) to report to the operator and receive orders and data from him, (b) to receive projectile and target data, (c) to aid the sensors, and finally, but not least, (d) to order the proper position to the gun.

Each of the eight peripheral devices is now described from a functional viewpoint. The type of transfer is also indicated; whether it be just input, just output, or both and whether the transfer involves numeric data (words) or pushbutton status (bit) data. The numbering system used here is the same throughout the remainder

TABLE 8-I. AFAADS COMPUTER PERIPHERALS

<u>INTERFACING EQUIPMENTS</u>	<u>DIGITAL PERIPHERAL DEVICES</u>
Operator's Panel	D1. Initialization Data, Operator's Panel
	D2. Control and Status, Operator's Panel
	D8. Test Mode, Operator's Panel
Tracking Head	D3. Sensors
	D4. Rate Gyro
	D6. Sensor Servos
Gun Mount	D5. Gun Servos
Vehicle Orientation Sensors	D7. Vehicle Orientation Sensors

of this volume.

D1 Initialization Data, Operator's Panel: (Input and output, word and status (bit) data)

D1.a Inputs. Operator entered initialization parameters and their values. The initialization parameters are air temperature, air pressure, relative wind direction, wind velocity, muzzle velocity, dive angle threshold, and constant turn (acceleration) threshold.

D1.b Outputs. (1) The previously stored values for these parameters and (2) the redisplay of newly entered values as a check that they have been correctly entered.

D2 Control and Status, Operator's Panel: (Input and output, status (bit) data)

D2.a Inputs. Operator designated operational modes and status data relative to an engagement. The AFAADS operating modes are Initialization, Standby, Semi-Automatic Fire Control, Manual Fire Control, and Test. Fire control status data are the projectile type (one of five) and the desired types of target track prediction.

D2.b Outputs. Operating mode of the AFAADS gun fire system, projectile type being used in the ballistic computations, types of target track prediction being used, any losses in tracking sensor data, begin fire/cease fire indication, and computer not operating alarm.

D3 Sensors: (Input, word data)

D3.a Target Data. Laser output slant range  $D(j)$  and FLIR output of angular position of target in azimuth  $A(j)$  and elevation  $E(j)$  at sampling time  $t(j)$ .

- D3.b Projectile Miss Data. Laser output of projectile-to-target range difference,  $\Delta D(j)$ , and FLIR output of projectile angular miss in transverse angle,  $\Delta T(j)$ , and elevation angle,  $\Delta E(j)$ , at time interval  $t(j)$ .
- D4 Rate Gyros: (Input, word data) Angular rates of the gyro stabilized, FLIR/laser sensor head in transverse,  $\dot{T}(j)$ , and elevation,  $\dot{E}(j)$ , coordinates at time interval  $t(j)$ .
- D5 Gun Servos: (Output, word data) Gun lead angle data relative to sensor head in azimuth,  $\delta(j)$ , and elevation,  $\sigma(j)$ , at end of processing interval  $t(j)$ . These are the key outputs of the fire control computer program and represent the proper azimuth correction and elevation correction for the given target, prediction mode, and projectile characteristics.
- D6 Sensor Servos: (Output, word data) Sensor regenerative drive angular rates in transverse,  $\dot{T}(j)$ , and elevation,  $\dot{E}(j)$ , coordinates and range value,  $D(j)$ , for time period  $t(j)$ . These values override sensor internally generated values when the target is behind an obstacle or during momentary target fades. This function could be expanded to include sensor drive signals under high angular acceleration conditions in order to minimize the sensor from "getting behind".
- D7 Vehicle Orientation Sensors: (Input, word data) Vehicle pitch and cant from the vehicle orientation sensors.
- D8 Test Mode, Operator's Panel: (Input and output, word and status data)
- D8.a Input. Operator commanded phase of system test, test initialization data, and request for test results.
- D8.b Output. Computer feedback to verify commands, automatic

presentation of certain test results, and test status.

The interface of each of these peripheral devices with the digital computer is illustrated in Figure 8-1.

## 8.2 SPECIFIC INTERFACE SIGNALS

With reference to Figure 8-1 and the above discussion, the specific signals between each of the eight peripheral devices and the computer are listed. For the three peripheral devices that are a part of the Operator's Panel, reference should also be made to Figure 7-1.

### D1 Initialization Data, Operator's Panel

#### D1.a Inputs.

##### 1) Initialization Parameter. Status bit for one of:

AIR TEMP (Air Temperature)

AIR PRESS (Air Pressure)

REL WIND DIR (Relative Wind Direction)

WIND VEL (Wind Velocity)

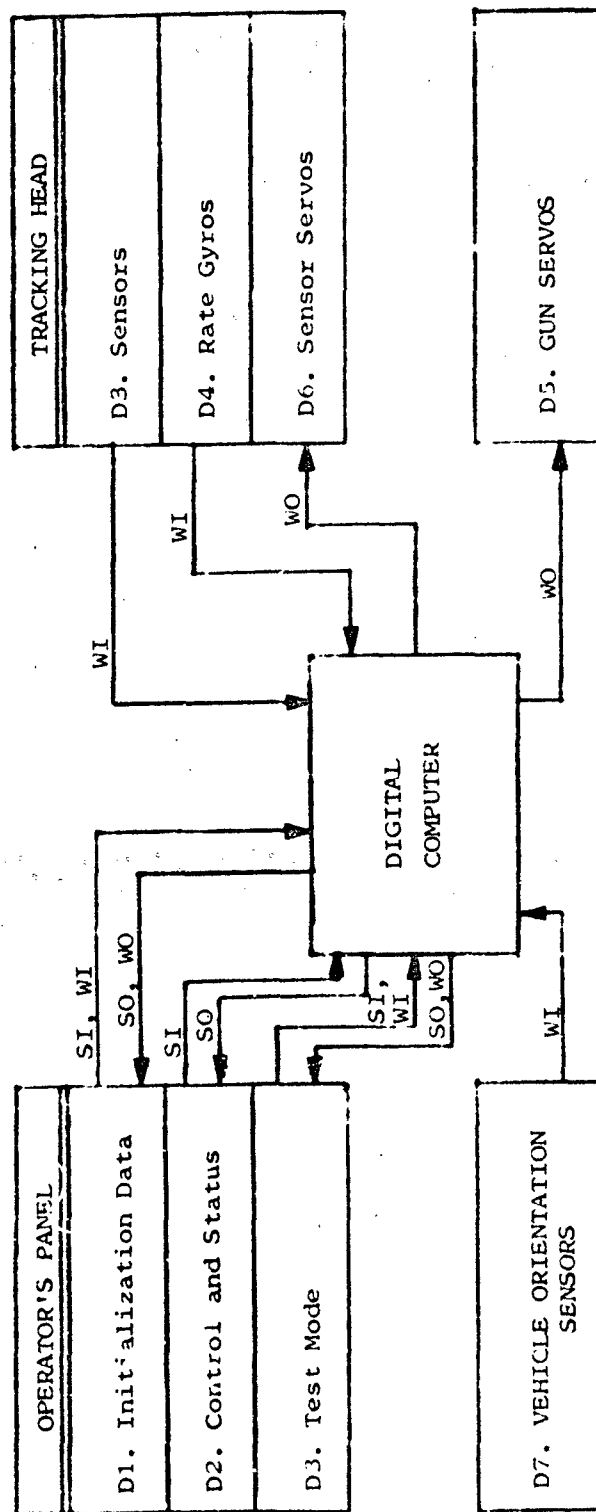
MUZZLE VEL (Muzzle Velocity)

DIVE THRESH (Dive Threshold)

TURN THRESH (Turn Acceleration Threshold)

Activation of one of these buttons must be preceded by the BEGIN INIT I mode button of peripheral device D2.

2) Value. Status bit if preset value is entered or word data (up to seven digits plus algebraic sign) if numeric entry from keyboard. Preset value is entered by pushing PRESET for the activated Initialization Parameter. Word data is entered by first entering value on the keyboard and then pressing ENTER. (Sec. 7.2 for detailed procedures).



Key: SI Status (Bit) Input  
 SO Status (Bit) Output  
 WI Word Input  
 WO Word Output

Figure 8-1. Input/Output Diagram

**D1.b Outputs.**

1) Stored Values. Upon activation of an Initialization Parameter switch, the computer will send to the Operator's Pane;;:

- a) Status Bit: To light designated Initialization Parameter switch indicator.
- b) Status Bit: To light OUT DISPL (Output Display) indicator on the keyboard.
- c) Data Word (up to seven digits plus algebraic sign):  
Value of the Initialization Parameter for display on the LED readout.  
Note, either RESET or USER has already been lighted.

2) Redisplayed Values. In response to the entering of a value for the Initialization Parameter, i.e., pressing PRESET or ENTER, the computer will send:

- a) Status Bit: To light PRESET if this key was pressed or USER if a value was entered on the keyboard.
- b) Status Bit: To light INPT DISPL (Input Display) indicator on the keyboard.
- c) Data Word (up to seven digits plus algebraic sign):  
Value of the Initialization Parameter for display on LED readout. This is either the Preset value or the just entered value.

## D2 Control and Status, Operator's Panel

### D2.a Inputs.

- 1) Operational Mode. Status bit in response to operator selection of a desired mode:

BEGIN INIT I (Initialization I Mode)  
STBY II (Standby II Mode)  
MANUAL (Manual Fire Control Mode)  
SCAN (Semi-Automatic Fire Control Mode)  
BEGIN TEST (Test Mode)  
BEGIN NEW TARGET (Semi-Automatic Fire Control Mode for a new target).

- 2) Projectile Type. Status bit indicating selection of one of five:

A HE (Armor Piercing High Explosive)  
A T (Armor Piercing Tracer)  
HE AT (High Explosive Anti-tank)  
TEST (Test Round)  
SUBCAL (Subcaliber High Velocity, High Explosive Round).

- 3) Prediction Modes. Status bits indication activation or deactivation of each of:

USE LINEAR (Use linear prediction only)  
USE DIVE (Use energy conservation prediction for a diving attack)  
USE TURN (Use constant turn rate prediction)

### D2.b Outputs.

- 1) Operating Mode. Status bits to light the indicator to show the operating mode the computer is in, a feedback to operator button action, except as noted.

STBY I (lights when power is first turned on)  
BEGIN INIT I  
STBY II  
MANUAL  
SCAN (lights in response to either SCAN or  
BEGIN NEW TARGET button action)  
BEGIN TEST

- 2) Projectile Type. (a) Feedback status bit to light the selected projectile type and (b) data word (four digits) to indicate the associated muzzle velocity on the LED display.
- 3) Prediction Mode Feedback. Feedback status bits to light or extinguish the appropriate indicators as a result of operator action in prediction mode selection.
- 4) Engagement Status. Status bits to indicate the progress of a Semi-Automatic Fire Control Mode attack. As the attack progresses, the following indicators will be successively lit: (See scenario of Section 7.2)

SCAN (In response to operator action as indicated above.)

ACQ (Angle tracking servo activated, laser activated and detects target. Target data is being acquired by the computer and the tracking filter is starting to fill. One second of data is required by the computer)

TRACK (Tracking filter has at least one second of data and the fire control solution is being generated)

In addition, if all three Prediction Mode switches are deactivated (see Peripheral Device D2.a.3 above), the target prediction algorithm in the fire control

solution will activate one or more of the following indicators:

AUTO LINEAR (Always on)

AUTO DIVE (Dive angle exceeds threshold)

AUTO TURN (Turn acceleration exceeds threshold)

5) Weapon Firing Indication. Status bits generated by the fire control computations to light:

FIRE BEGIN (Signifies target is in range and a fire control solution exists, firing can begin.)

CEASE FIRE (Signifies target is going out of range)

6) Tracking Status. Status bits generated by the target tracking and projectile miss sensing algorithms:

ERROR LOST TARGET (No target tracking data has been received for two seconds)

REGEN POSITION (Computer is regenerating the target position to drive the sensor head, results from target behind an obstacle or fading)

ANGLE DATA LOST (FLIR is not providing angle data)

RANGE DATA LOST (Laser is not providing range data)

PROJ SENS (Projectile miss distances are being sensed by the FLIR and laser)

ANGLE RATE HIGH (The slew rates on the sensor are too high for gathering useful projectile miss data)

7) Computer Inoperative Alarm. Status bit to activate ALARM indicator that the computer or one of the peripherals is not operating correctly. Designed to function even if the computer is inoperative. (See Sec.10)

### D3 Sensors

Total of six data word inputs to the computer as given below. The computer word length assumes rounding to the LSB rather than truncating.

D3.a.1 Target Angle Data: Two word data input, azimuth and elevation to target from FLIR, 15 accurate unsigned bits in azimuth and 13 accurate bits and sign bit in elevation out of sixteen bit data words.

D3.a.2 Target Range Data: One word data input, slant range to target; accuracy of 16 bits in sixteen bit word. No sign bit needed.

D3.b.1 Projectile Miss Angle Data: Two word data input, transverse (line-of-sight) angle and elevation angle between mean position of all observed projectiles in the field of view (FOV) of the FLIR and the target. Being a difference,  $\Delta E$  is accurate to 8 bits plus one of sign,  $\Delta T$  is accurate to 9 bits plus sign. 16 bit words.

D3.b.2 Projectile Miss Range Data: One word data input; difference is slant range between the mean range of all observed projectiles and target, as measured by the laser. Accurate to 12 bits plus sign. 16 bit word.

#### D4 Rate Gyros

Input slew rates of gyro in transverse and elevation angular coordinates.  $\dot{T}$  and  $\dot{E}$  are measured. These figures will match state of the art unclassified devices yielding accuracies of 0.6%. Transverse rate should have eight bits plus sign bit. Elevation rate should be eight bits plus sign bit. Standard 16 bit data words are assumed.

#### D5 Gun Servos

Gun aim lead angles generated by the computer and sent to the gun servos. The ideal case would have the gun aimed with accuracies approaching the FLIR and laser input data. Geometry and ballistic solution accuracies yield these approximate values:

13 bits plus sign for elevation; 13 bits plus sign for azimuth. 16 bit words.

D6 Sensor Servos

Sensor servos fed computed angular rates and range position if FLIR and/or laser target data are not available. Rates are in transverse and elevation rate coordinates. Rate accuracies at best will match D4, i.e., transverse rate to eight bits plus sign; elevation rate to eight bits plus sign. Range accuracy may match original range data, namely 16 bits without sign. Use sixteen bit words.

D7 Vehicle Orientation Sensors

Data on the vehicle pitch and cant are read automatically at the beginning of a Semi-Automatic Fire Control Mode mission. Each is accurate to six bits (six-tenths of a degree) plus sign bit in a sixteen bit word.

D8 Test Mode, Operator's Panel

D8.a Inputs.

1) Phase of System Test. Operator action resulting in status bits to indicate the desired phase of a system test.

INIT SYSTEM TEST (Button action to enter the initialization phase of system test, i.e., to enter system test initialization data)

BEGIN SYSTEM TEST (Action to initiate the simulated target track and the AFAADS dynamic system test)

END TEST (Action to stop the dynamic system test short of the standard 16-second test period)

2) Test Initialization Parameters. Status bits indicating which dynamic system test initialization parameter is being entered.

TARGET FROM LEFT or

TARGET FROM RIGHT (Indicates direction simulated target will pass the gun)

ENTER E (Preparatory to entering the initial target elevation angle on the keyboard)

ENTER A (Preparatory to entering the initial target azimuth angle on the keyboard).

3) Test Initialization Value. Numeric entry from the keyboard on the initial elevation or azimuth angle for the simulated target. This action must be preceded by either ENTER E or ENTER A button actions. Input will be data words representing up to three digits or up to two digits plus algebraic sign.

4) Test Results. Status bits indicating request for numeric readout of (only one of):

NO OF PROJECT (Number of projectiles on which miss data was recorded)

ELEV BIAS (Mean elevation miss angle for all observed projectiles)

AZM BIAS (Mean azimuth miss angle for all observed projectiles)

#### D8.b Outputs.

1) Phase of System Test. Actual status of system test, either as a response to operator action or on a time basis. Appropriate indicator will light.

INIT SYSTEM TEST (In response to operator action)

BEGIN SYSTEM TEST (In response to operator action)

GOOD, FAIR, or POOR (Overall evaluation of system test, activated either 16 seconds after BEGIN

SYSTEM TEST button action or after END TEST button action).

2) Test Initialization Parameter Feedback. Feedback status bits to light the appropriate Test Initialization Parameter indicators.

- 3) Test Initialization Data Feedback. Feedback for display of initialization data entry on the LED display. Also status bit to light INPT DISPL (Input Display).
- 4) Test Results. Numeric value of requested data results for presentation on LED display plus status bit to light appropriate indicator.

## SECTION 9

### SOFTWARE

#### 9.1 INTRODUCTION

This section describes the software required for the AFAADS computer (Subsection 9.2), its requirements on the computer (Subsection 9.3), and how the software may be produced (Subsection 9.4). The software, as covered in Subsection 9.2, is treated in three categories: Software which controls the operation of the AFAADS computer (executive), software which performs the AFAADS functions (applications software), and software which assures that the computer and its parts are still working (test software).

An analysis is presented of the applications software to show the real time processing requirements of the computer and to form a basis for estimating the cycle time and memory requirements for the processor.

Finally, a discussion of software production results in recommended languages, implementation techniques, and supporting equipment required to build the software system.

#### 9.2 SOFTWARE PARTITIONING

As stated above, the software can be divided into three categories. Each is now discussed.

##### 9.2.1 Control Software

The control software performs the executive functions for the AFAADS gun system. The primary objective of this set of programs is to monitor and control the interaction of operational and test software with the various hardware devices of the system, as shown in Figure 9-1.

System operation is controlled by two types of interrupts: timer interrupts and fault-indication interrupts. Timer interrupts are the major controlling function of the system; they poll all input devices including the Operator's Panel at specified periodic intervals. When the Gunner/Operator makes an entry, it is received by software at the

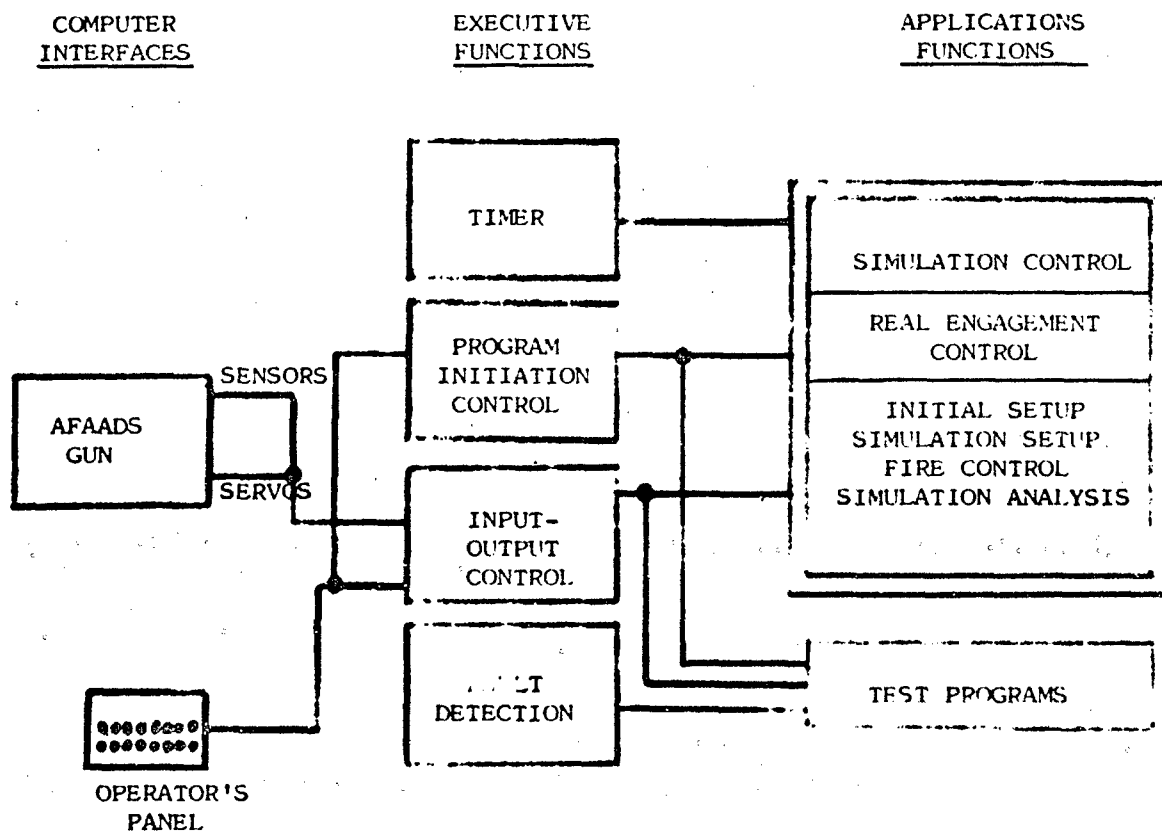


Figure 9-1. Executive Functions

next polling interval. An entry from any peripheral device initiates specified software programs or may provide the necessary data for a suspended program to continue. Data values entered are passed on to operational or test programs, according to the type of action requested by the user. If software initiation is required in response to a user button selection, currently operating functions are first completed and then a control function initiates the proper program.

Fault-indication interrupts cause the initiation of a fault detection program. This program determines the problem (e.g., power failure, program failure, etc.) and may either return control to the interrupted program, initiate test action, or shut down the computer. Failures are also indicated to the Operator through the Alarm indicator as discussed in Section 8.2.

An Input-Output Control Program (IOCP) performs the interfacing functions required to poll and receive input data and to format and send output data as requested. The IOCP consists of device handlers for the Operator's Panel, the servo devices, and the sensors.

Input handlers are to be provided for the Operator's Panel, laser, FLIR, rate gyros, and vehicle orientation sensors. These are all polled for input at specified periodic intervals (e.g., 0.1 seconds). Output handlers will be provided for the Operator's Panel (indicator lights and numeric readouts), and to the gun and sensor servos. Output activity is performed upon demand by some other program, rather than being timer driven. The input handlers accept data in ASCII or binary notation and convert it to properly-scaled binary values for applications processing. The output handlers scale and convert output binary data to the proper device codes.

#### 9.2.2 Applications Software

The applications software suit consists of two sets; real engagement software and simulated engagement software. The interrelationships of applications software modules are shown in Figure 9-2.

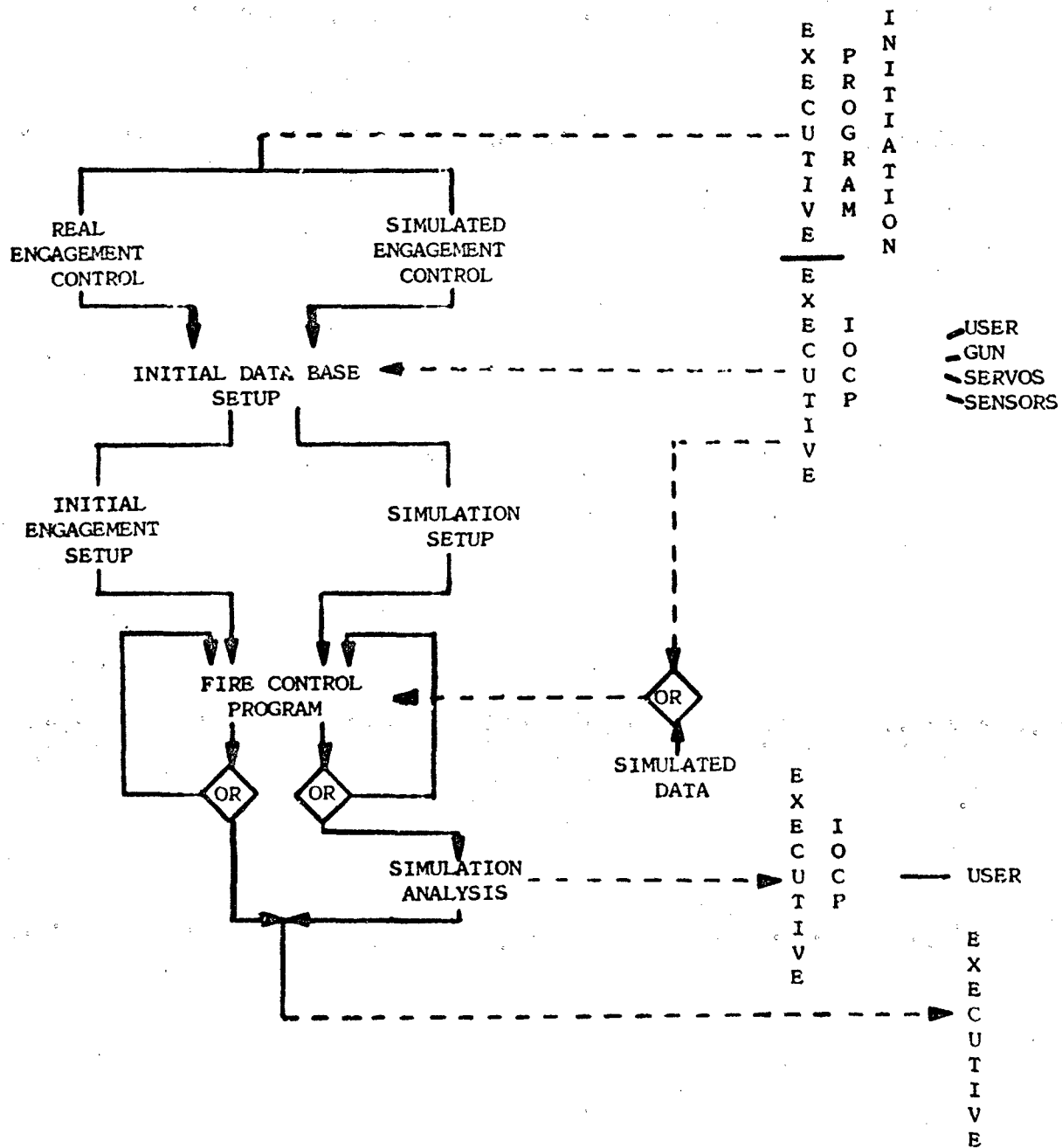


Figure 9-2. Applications Software Modules

#### 9.2.2.1 Engagement Control Program

The engagement control program sets up the data base for real time operations and controls the semi-automatic fire control program (FCP) in real time. Whenever a new engagement is begun, the Engagement Data Setup Function is called. Then the FCP is entered, with parameters to show that an actual, rather than a simulated, engagement is desired.

#### 9.2.2.2 Simulation Control Program

Whenever a system test is requested, i.e., the Operator places the AFAADS in the Test Mode, the program first performs extensive diagnostic test programs to check the computer (the Confidence Self Tests described in Section 13.2). This is normally followed by the Dynamic System Test, initiated when the Operator presses INIT SYSTEM TEST (Initiate System Test). The simulation control program then sets up the data base for the Dynamic System Test of the combination of fire control system and the entire weapon system (described in Section 13). The FCP is now entered under control of the simulation control program and with parameters to show that a simulated, rather than an actual, engagement is in progress.

The simulation control program manages the data describing the simulated flights engaged. It also maintains system test indicators, a measurement of effectiveness of the test based on projectile miss data.

At the conclusion of the Dynamic System Test (typically after 16 seconds of simulated engagement), the measures of effectiveness are statistically reduced and presented to the operator along with a Good/Fair/Poor indicator of results. The Operator can also request numeric readouts on the results.

#### 9.2.2.3 Initial Data Base Setup

The initial data base setup function allows the user to provide data base entries of slowly varying or constant items. The function receives control from the executive program initiator when the operator depresses BEGIN INIT I (Begin Initialization I). The data base will contain the last preset or user-entered values. The program then allows

the user to select and examine/change all specified data items. The selected data items will be automatically scaled or converted between user and binary notations by the executive IOCP.

The data base items accessible to the user, in this mode, include:

- o Air temperature
- o Air pressure
- o Dive angle threshold
- o Turn (accelerator) threshold
- o Relative wind direction
- o Wind velocity
- o Muzzle velocity

#### 9.2.2.4 Engagement Data Setup

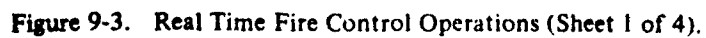
The initial data base setup program is then entered for each test and each engagement, to compute coefficients for the real-time ballistics computation equation.

Input to this program is provided by the Operator's Panel handler, and the vehicle orientation sensors handler. The Operator's Panel handler determines user selections of projectile type and muzzle velocity; the vehicle orientation sensors provide cant and pitch. The program then computes parameters for the time of flight equation and vehicle orientation data.

#### 9.2.2.5 Fire Control Program (Real time and simulated operations)

The Fire Control Program (FCP) operates under either the real engagement control program or the simulation control program. Figure 9-3 shows the overall processing performed by the FCP.

The FCP program is entered cyclically at pre-specified time points (every 0.1 seconds), until the end of the simulated or real engagement is indicated. Each cyclic execution of the program ends with a wait for the next specified timer interrupt point to occur. At that time, a timer interrupt via the executive will begin the next cycle. The FCP program, in determining target and projectile data, either generates



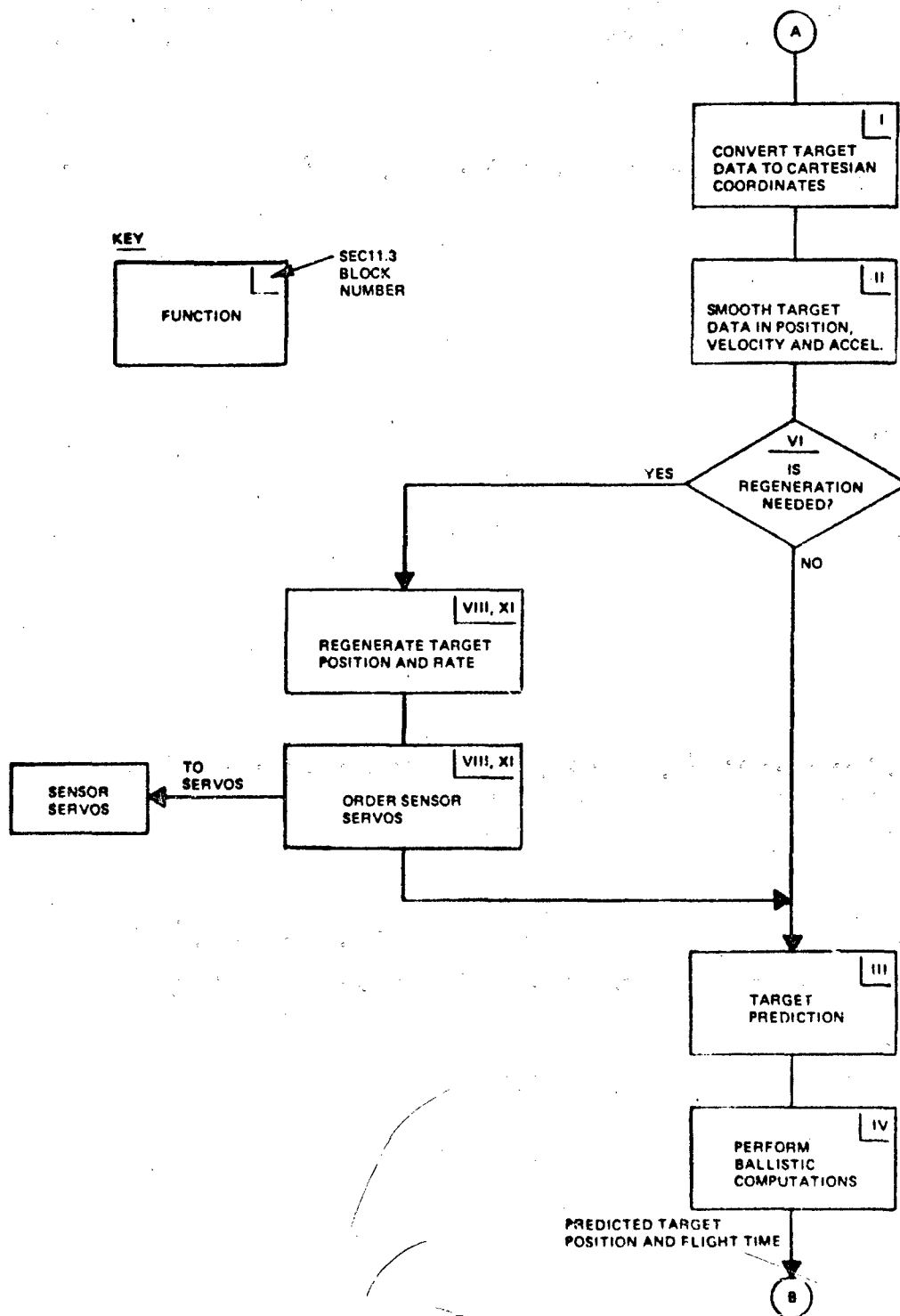


Figure 9-3. Real Time Fire Control Operations (Sheet 2 of 4).

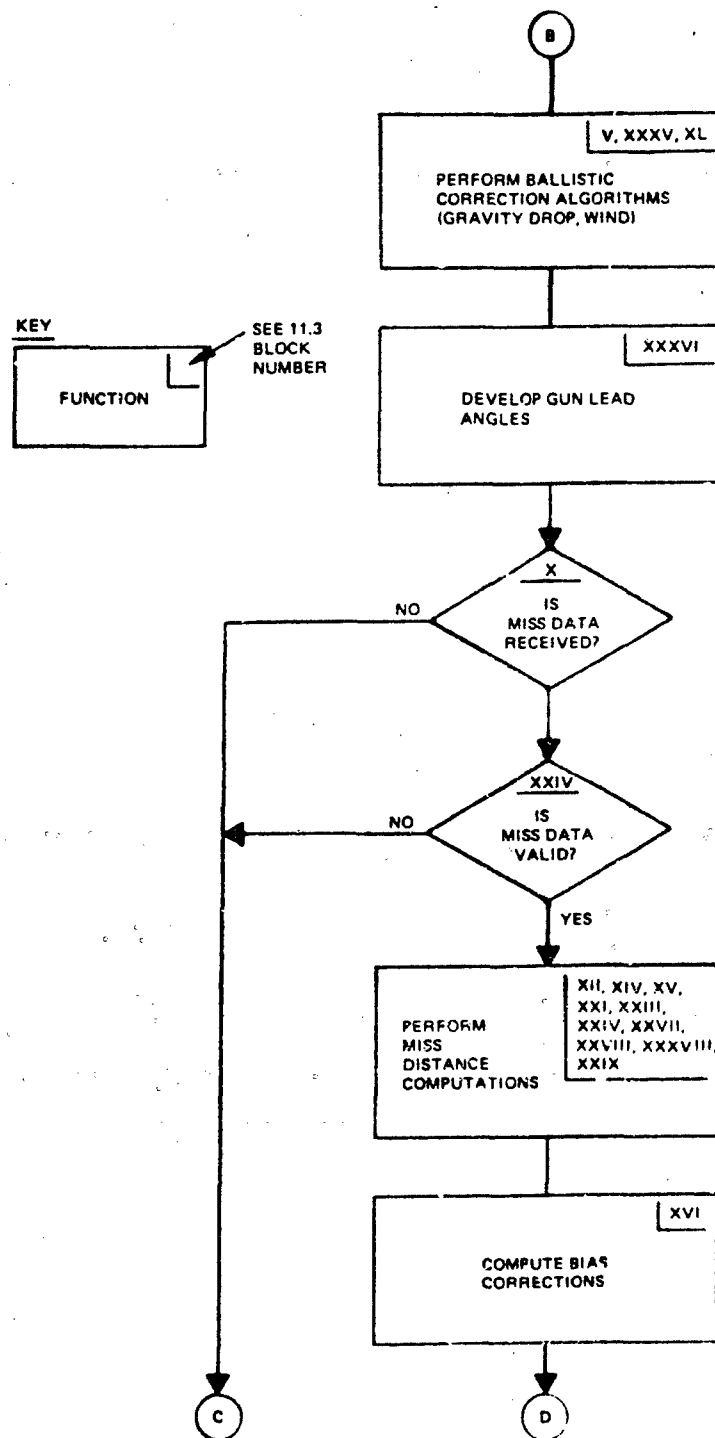


Figure 9-3. Real Time Fire Control Operations (Sheet 3 of 4).

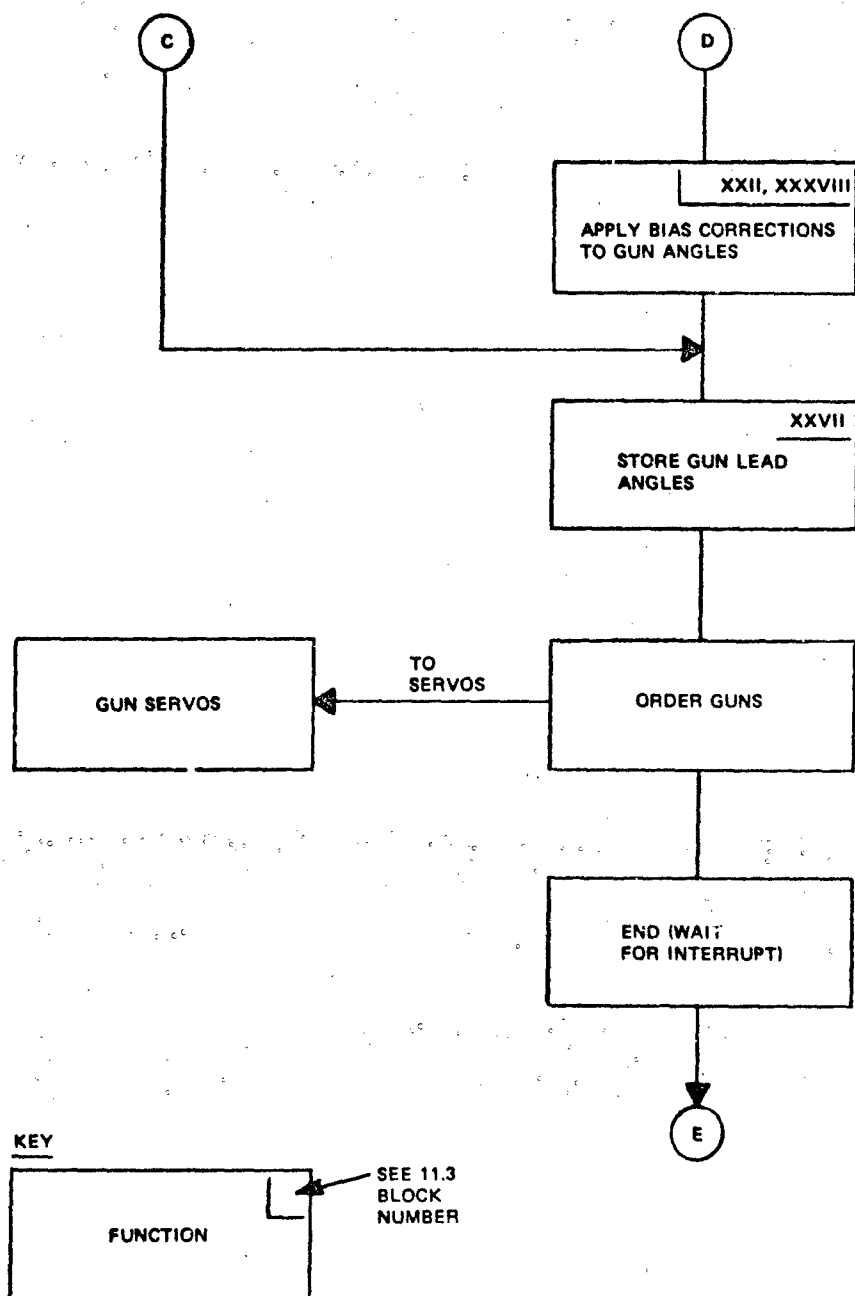


Figure 9-3. Real Time Fire Control Operations (Sheet 4 of 4).

it when under simulation control, or reads sensors under real engagement control.

In the following discussion, detailed program description cross references are provided by Roman numerals in parenthesis. These reference the blocks of Figure 9-3 and Section 11.3, block descriptions.

#### 9.2.2.6 Program Cycle Overview

These activities are reinitiated cyclically at pre-specified time points (every 0.1 sec). First, target position data is inputted from the sensors (VI), converted to Cartesian coordinates (I) and smoothed (II). If regeneration of target data is required (target range D and/or transverse and elevation angular velocities  $\dot{T}$ ,  $\dot{E}$ ), it is computed (VIII and XI) and outputted to the tracking sensor (sensor I/O handler).

Impact point prediction (III) and ballistics computations (time of flight) (IV) are performed to give projectile time of flight and predicted impact point. Gun lead angles are generated by the following operation: Coordinate conversion (vehicle to earth) (XL), ballistics corrections (V), coordinate conversion back again (earth to vehicle) (XXXV), and gun lead angle generation (XXXVI).

The miss data is received (X) and is validated (XXIV); miss data computations are initiated (XXVII). The ballistic correction (XII) computes the time of flight to reach the current target position. The time of projectile fire (XXXIX) is matched against stored trajectories (XIV) to compute the predicted target azimuth and elevation (XV). This provides the basis for a target maneuver correction. It is calculated (XXVIII) and converted to a line-of-sight transverse correction angle (XXIII). (Detailed analytic basis for these processes and those described in the next two paragraphs can be found in Volume I, Section 3 and in Reference 1 and 2.)

The reordered miss angles are determined (XXXVIII) as the difference between measured closed loop miss distance and that caused by target maneuvers and track prediction errors.

The bias correction algorithm (XVI) then allocates the residual bias errors into bias corrections on muzzle velocity, gun azimuth and gun elevation. The bias muzzle velocity correction factor is then applied (XXII) to the gun's azimuth and elevation lead angles. The lead angles are then corrected for azimuth and elevation biases (XXXVII) and outputted to the gun mount (gun I/O handler). The FCP then waits for the next 0.1 second timer notification from the executive before cycling to the beginning of the FCP program, as described above.

### 9.2.3 Software Test Program

Three types of self-test programs are provided. These test programs are static tests of the data processing hardware items, as opposed to the dynamic system test afforded by the simulation control program. The self-tests are tests of (1) the processor, (2) the memory, and (3) the I/O portions of the computer. The breakdown of the self-test functions is shown in Figure 9-4. Each is now described.

The processor test program performs a test of arithmetic capabilities, logic capabilities, control/interrupt capabilities, and fault detection features. The test program is arranged so that a limited subset of it can be exercised in real time (automatic self-test) or the entire program can be executed repeatedly for a few seconds when requested by the operator (confidence self-test).

The memory test program tests the readable portions of PROM (programmed read only memory) storage by computing their checksums at least two different ways, and checks core (dynamically changeable) storage by computing an initial checksum of existing data, and performing worst pattern complementary and address checks on small portions of core. While a portion of core is being tested, its contents are moved elsewhere for the duration of the test. At the completion of the core test, a final checksum of the data is compared to the initial checksum to insure that the data has not been altered by the test itself.

The I/O test program checks the input/output functions of the computer to assure their operation. Both internal data and control paths, and also external (cabled) paths will be tested by this function.

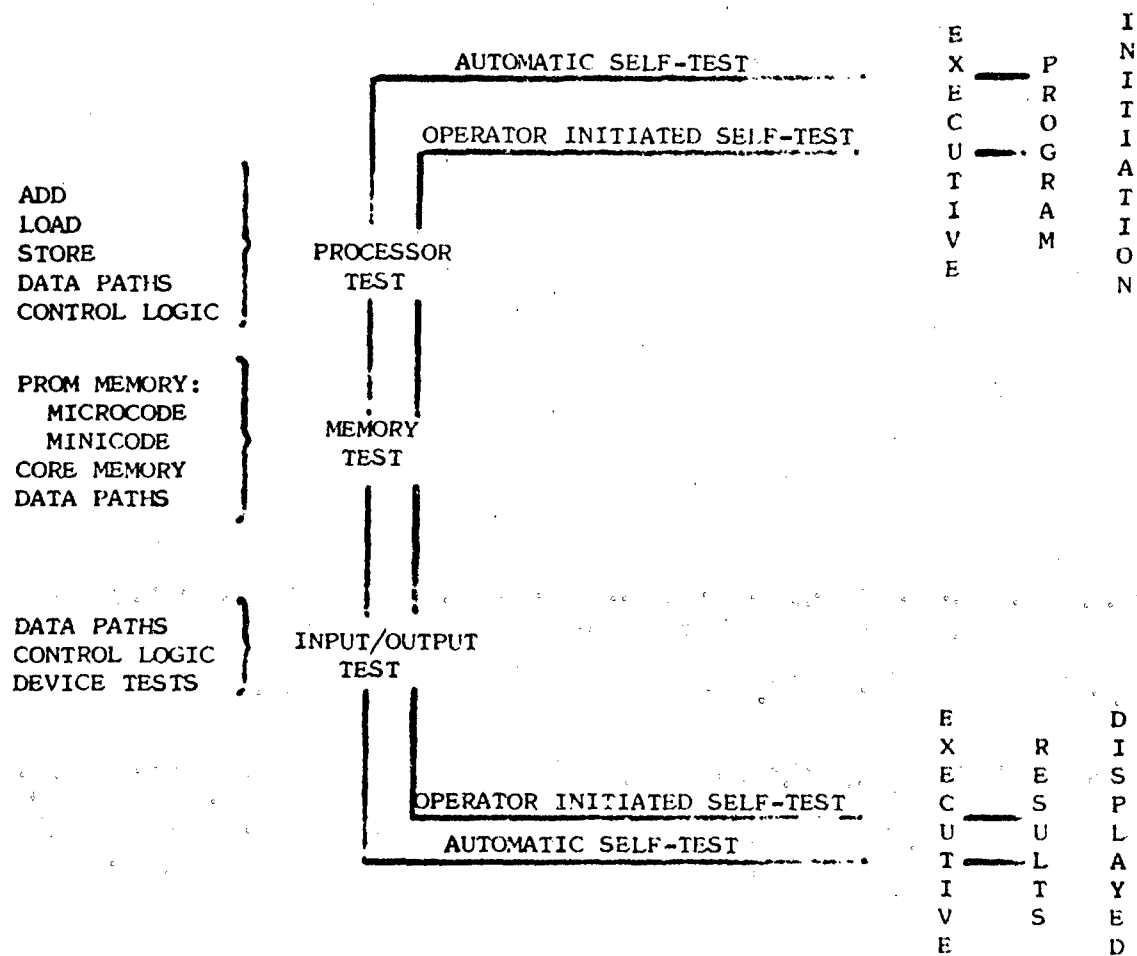


Figure 9-4. Test Functions

### 9.3 ANALYSIS OF APPLICATIONS REQUIREMENTS

The above described proposed applications software design is now analyzed for two purposes:

- a. To determine processing capacity requirements of the computer and hence the required machine speed.
- b. To determine storage capacity requirements for code and data.

Only operational real-time software is considered in the processing capacity requirements study; however, the full suit of software is considered in the storage capacity requirements analysis.

#### 9.3.1 Processing Requirements

Processing capacity requirements are analyzed to determine the required speed of the computer for a given set of instruction mixes. The required speed serves as a dual indication of: (1) the class of computing equipment required for the job and (2) the reasonableness of algorithms used for this job.

The types of instruction sets available on computers have a significant effect on a computer's processing speed for a given application. They also affect the amount of storage used by program code.

This application is designed to be suitable for a militarized equivalent of today's mini-computers. Such equipment, as the processor capacity results show, is adequate but not excessive for the AFAADS requirements.

Engineering experience in the use of such mini-computer equipment has led to reliable means of judging processing loads for well-defined computer functions. The fire control program, for real time operations, is sufficiently well defined in this concepts analysis for the development of detailed estimates of code. These estimates were then used to determine the number of typical machine operations. This is a deterministic load, where iteration factors and algorithmic processing load are specified by design; and thus an accurate determination of processing load can be made.

In estimating code for the real time program operations, three types of program code units are considered. They are:

- (a) Memory to register (simple Arithmetic/Logic Unit - ALU) operations, typically add, subtract, compare, load and store. These operations have the same difficulty level in all mini-computers of recent design (older design computers can also be accommodated in this scheme, as will be shown later).
- (b) Multi-step ALU/micro-code operations, such as multiply and divide. These operations are sufficiently prevalent in the AFAADS real time processing to require their separate tallying, especially since most processors perform Type (a) instructions with relatively equal speed, but multiply/divide instructions typically take somewhat to much longer and vary more from machine to machine. Furthermore, multiply/divide instructions may be either microprogrammed or simulated by subroutine, further requiring their separate tally.
- (c) Complex trigonometric functional operations. Trigonometric functions used in the programs were tallied in terms of equivalent sine or cosine function references. These functions may all be performed by software or may, in cases of user microprogrammable mini-computers, be performed by firmware. These functions occur sufficiently often to warrant there being separately tallied.

Operations which are not of Types (a), (b), or (c) above are analyzed as being some mixture of (a), (b), and (c) -type operations, following standard programming practices and standard (published) algorithms, when available.

Each operation's operands are analyzed to determine whether 16 bits of precision are adequate. When greater precision is required, as for intermediate unscaled results, doubleword operations are employed (on trial coding). These are counted as 1.5 regular Type (a) instructions for

speed determination (three memory cycles where two are normally adequate). It is assumed that double basic arithmetic and shift instructions will be microcoded.

For nearly all final data results, after scaling, 16 bits of storage are adequate. (Details on storage needs are indicated in Section 11.3 where relevant.)

Based on the type of instruction set utilized, as described above, and using mini-computer microcoding experience, a PROM microcode memory of 1000 36- to 48- bit microcode words appears adequate for the desired instruction set (excluding trigonometric microroutines).

Processing capacity and speed are analyzed by examining the processing to be performed by each system block (reference Section 11.3), summing the results and multiplying by the expected rate of invocation of the block. Table 9-1 summarizes the results. The columns of the figure represent:

- o Block number is the operational block cross referenced to Section 11.3.
- o Description covers the task performed by that block.
- o Extended load is the estimated processing load per unit frequency in terms of:
  - a. Estimated quantity of memory-reference-type instructions normalized to load/add/store type units.
  - b. Estimated quantity of multiply or divide instructions which may be performed by hardware (three load/add/store units) or by software simulation.
  - c. Estimated quantity of trigonometric functions, normalized to sine/cosine difficulty, which may be performed by hardware microcode or by software subroutines.
- o Composite load is a summarized load based upon an equivalent rate per second for each of the three load units. Specifically, the composite load is summarized for a hypothetical

TABLE 9-1 AFAADS REAL TIME PROCESSING LOAD (Sheet 1 of 3)

BLOCK (Sec. 11.3)	DESCRIPTION	EXTENDED LOAD/SEC			COMPOSITE LOAD/SEC	
		LOAD/ADD/ STORE UNITS	MULT/DIVIDE UNITS	TRIG UNITS	EQUIVALENT UNITS	% TOTAL
I	Conversion to Cartesian	1,100	280	4	2,220	2.04
II	Target Tracking & Smoothing	7,000	630	0	8,890	8.16
III & IV	Impact Point Prediction	7,000	1,900	0	12,700	11.66
V	Ballistic Corrections	110	220	90	7,070	6.49
VI	Missed Data Points Search	900	0	0	900	0.83
VII	Missed Target Data	200	0	0	200	0.19
VIII & XI	Regenerative Target Position and Rate	620	160	30	3,200	2.94
X	Receive Miss Data	5,050	0	0	5,050	4.64
XII	Ballistics Correction (Block IV but no iterations)	150	40	0	270	0.25
XIII	Ballistics Data Store	1,000	0	0	1,000	0.92
XIV	Match Times	2,100	30	0	2,190	2.01
XV	Extract Angles	240	60	0	420	0.39
XVI	Bias Correction Algorithm	20,000	2,000	80	31,600	29.00
XVII	Stop Miss Distance Computation	5,000	0	0	5,000	4.59
XX	Coordinate Conversion to Polar	960	330	3	2,160	1.99

Continued...

TABLE 9-1 AFAADS REAL TIME PROCESSING LOAD (Sheet 2 of 3)

BLOCK (Sec. 11.3)	DESCRIPTION	EXTENDED LOAD/SEC			COMPOSITE LOAD/SEC	
		LOAD/ADD/ STORE UNITS	MULT/DIVIDE UNITS	TRIG UNITS	EQUIVALENT UNITS	% TOTAL
XXI	Read Angle Extraction	2,000	400	0	3,200	2.54
XXII	Bias Muzzle Velocity Correction	400	70	2	750	0.69
XXIII	Coordinate Conversion to LOS	1,500	30	1	1,660	1.52
XXIV	Miss Distance Measure- ment Validity	600	40	0	720	0.66
XXV	Total LOS Angular Velocity	600	300	0	1,500	1.38
XXVI	Coordinate Conversion to Polar	600	200	4	1,480	1.36
XXVII	Initiate Projectile Miss Calculation	100	0	0	100	0.09
XXVIII	Target Maneuver Correction	500	80	0	740	0.08
XXIX	Real Time Clock & Exec O/H	1,000	0	0	1,000	0.92
XXX	Ballistic Data Input Register	120	0	0	120	0.11
XXXV	Coordinate Conversion (earth to vehicle)	750	250	150	12,000	11.01
XXXVI	Gun Lead Angles	100	0	0	100	0.09

Continued. .

TABLE 9-1. AFAADS REAL TIME PROCESSING LOAD (Sheet 3 of 3)

BLOCK (Sec. 11.3)	DESCRIPTION	EXTENDED LOAD/SEC			COMPOSITE LOAD/SEC	
		LOAD/ADD/ STORE UNITS	MULT/DIVIDE UNITS	TRIG UNITS	EQUIVALENT UNITS	% TOTAL
XXXVII	Azimuth & Elevation Bias Corrections	300	50	0	450	0.41
XXXVIII	Residual Miss	100	0	0	100	0.09
XXXIX	Time of Projectile Fire	200	0	0	200	0.18
XL	Coordinate Conversion (vehicle to earth)	800	200	8	1,960	1.80
		61,100	7,270	372	108,950	100.00

- NOTE: 1) Blocks IX, XVIII, XIX, XXXI, XXXII, XXXIII, and XXXIV involve Operator actions during the Initialization Mode, and hence are non-real time.
- 2) Composite load is defined as the sum of Load/Add/Store Units plus 3 times Multiply/Divide Units plus 70 times Trigonometric Units.

computer which performs multiplication/division operations in three load/add/store units and uses standard software algorithms for computing trigonometric functions, each in load/add/store units. The load is expressed in both equivalent load/add/store units and as a percent of the full load.

From the table we see that 108,950 composite load/add/store units must be executed each second, or 9 microseconds is the maximum allowable time per unit. Thus the average instruction cycle time for the computer must be less than 9 microseconds in order to insure complete real time processing each cycle. To provide an adequate safety margin, it is recommended that a 5 microsecond average instruction machine be used. This slow time is necessitated by the requirement for non-volatile core memory for the Operand Memory.

It should also be mentioned that if the trigonometric functions are performed by firmware rather than software, the instruction speed can be reduced. On the other hand, additional analysis is required on the input sampling frequency, currently set at 10 times per second. An increase in this rate will have a direct impact on the computer cycle time.

### 9.3.2 Memory Requirements

Memory requirements for program code were estimated in the estimating process for coding the blocks of the real time operations program (Fire Control Program), and by engineering judgment for the remaining real time and auxiliary functions. Memory requirements for program code were estimated in terms of the quantity of Type (a), (b), and (c) instructions combined, and an average size, in bits, for a "statistical" instruction.

Each Type (a) instruction typically represents 32 bits of code in most mini-computers when expressed as an instruction with full memory reference operand and a register operand for an arithmetic operation. However, experience has shown that, quite frequently, with good coding habits, about 30% of these instructions can be coded with both operands

in registers, which only takes 16 bits of storage. Thus, a Type (a) instruction was weighted as  $(0.30 \times 16 + 0.70 \times 32)$  bits = 27.2 bits (say 27 bits).

Each Type (b) instruction typically represents 32 bits of code in most mini-computers when expressed as an instruction where the multiplier and product are in registers and the multiplicand is a full memory reference operand. These instructions were counted as 32 bits of storage. This figure (32 bits) is also suitable for most machines which do not implement Type (c) instructions in hardware, for which a special "opcode" would generate a unimplemented hardware interrupt. The operating system fault control routine would then transfer to a subroutine which would perform the operation.

Type (c) instructions were treated as Type (b) instructions for purposes of program memory sizing.

With this as a basis typical blocks of the real time algorithms were analyzed with respect to the mixture of Type (a), (b), and (c) instructions. Based upon this, memory requirements for program minicode is estimated in Table 9-II. Since program code is serially reuseable and does not alter itself, it should reside in (programmable) read-only memory (PROM or ROM).

This takes care of the program. We now turn to the memory requirement for the constants and similar data. The memory requirements for data are estimated in Table 9-III. These estimates were made by a careful analysis of all data, constants and workspace required for program execution. Two requirements for data were considered: (1) data storage required for implementation of the algorithms of the programs; the size of such storage can be determined from the nature of the individual datums being analyzed and (2) data storage required for program control. This category includes register storage, subroutine calls, workspace stacks for multi-term expression development, and the like. This category of storage was structured, based on current software engineering practice. Data is broken down by type of storage required: (1) constant data must occupy read-only memory, (2) work storage data must occupy non-volatile

TABLE 9-II PROGRAM MINICODE SIZE

<u>PROGRAM</u>	<u>INSTRUCTIONS</u>	<u>16-BIT WORDS</u>
EXECUTIVE	600	1,000
APPLICATIONS PROGRAM:		
OPERATIONAL PROGRAM	6000	9,600
SIMULATION DRIVER	500	800
TEST PROGRAM	500	800
		12,200
SAFETY FACTOR 20%		2,440
		14,640

TABLE 9-III DATA (PARAMETER) MEMORY

<u>FUNCTION</u>	<u>ROM CONSTANTS</u>	<u>NONVOLATILE RAM</u>	<u>WORKSPACE RAM</u>
Executive Storage	40	64	28
Initialization, I/O	6	6	16
Standby II & Init II	75	0	26
Initialization for Engagement	0	0	0
Initialization for Test	6	8	3
Real Time Operations	200	200	300
Test Indicators	0	0	10
Test Program	50	10	50
	<hr/>	<hr/>	<hr/>
	377	288	1,453

random access (read/write) storage, and (3) work space storage must occupy read/write storage (can be volatile). Combining the results of the last two tables total memory requirements, exclusive of microcontrol memory, are summarized in Table 9-IV. A three level memory structure of read only, non-volatile read/write, and volatile read write is listed as well as a two level memory structure (where the last two are combined). The two level structure has been selected for the hardware, as presented in Section 10.

#### 9.3.3 Conclusion Summary

The conclusions from the previous analysis are:

- o The processor required by this job must have a typical add (memory to register) cycle faster than 9 microseconds (5 recommended), including instruction and data memory accesses.
- o Multiply/divide operations must be microcoded or hardwired.
- o Trigonometric functions may be minicoded if single (16 bit) word precision is adequate; otherwise microcoding is be required.
- o Word length of 16 bits will accommodate nearly all data items.
- o 32 bit simple arithmetic and 32 bit shift operations must be microcoded or hardwired.
- o 16K words of program/constant storage, and 2K words of core/RAM (Random Access Memory) storage would be adequate.
- o 1K of microcontrol memory is required for a basic instruction set with multiply/divide. Additional microcontrol memory may be required for trigonometric functions.

#### 9.4 PROGRAM PRODUCTION

Two types of program material may need to be produced for the AFAADS System: microcode (firmware) for special purpose functions and minicode (standard software material) for the bulk of the application programs, control programs, and test software. It is expected that, depending on the computer configuration chosen, microcode may represent from zero to 10% of the total effort.

TABLE 9-IV TOTAL MEMORY REQUIREMENTS

	<u>3-Level</u>			<u>2-Level</u>	
	ROM	RAM	WORK	ROM	RAM
Parameter Storage	377	228	1453	377	1741
Code	<u>14,640</u>	---	---	<u>14,640</u>	---
T O T A L	15,017	228	1453	15,017	1741

It is recommended that both microcode and minicode be implemented in a macro assembly type language utilizing current proven techniques of program production; namely, top-down program implementation, structured design and coding, complete documentation, and peer-level program design and code review.

Program development will require a Programming Support Center (PSC) to support both minicode and microcode assembly, linkage, loading, and debugging.

Field employment of the system will require access to a Programming Support Center for program maintenance and correction of latent defects (bugs) which may slip through final testing. Field depots, furthermore, require the capability for altering stored programs (such as by PROM replacement) and for servicing of the computer itself.

#### 9.4.1 Program Material

The partitioning of software functions between microcode and minicode depends on several factors which must be considered:

1. The processor chosen for the end-item configuration.
2. The capacity of the computer to perform the application programs based on varying allocations of program material to faster microcode or slower minicode.
3. Relative program production cost (nonrecurring) between microcode and minicode.
4. Relative recurring cost of providing micromemory versus minimemory.

Since Item 1 is not known at this time, only the following recommendations can be made. If the end-item computer is not user microprogrammable, then this discussion is academic.

If it is user microprogrammable, then the Programming Support Center must provide a means of assembling, debugging, and correcting microcode. Furthermore, if top-down programming management and development are followed, decisions on which routines are micro-implemented versus mini-implemented can be postponed to the latest moment in design, when

the maximum information on program sizing and the like is known.

Initial recommendations based on engineering judgement, are that trigonometric functions be microcoded due to their frequent usage.

#### 9.4.2 Language of Implementation

It is recommended that program code, both micro-implementation and mini-implementation be produced in a macro assembly language.

The use of an assembly language permits tight control of machine resources required to execute programs; namely, memory space and execution time. This factor is vital to sizing the AFAADS job for a highly mobile field-employable system.

The use of a macro capability in the assembly language facilitates the use of structured program techniques by permitting the following basic facilities:

- o CALL macros, to facilitate inter-subroutine linkage with parameters.
- o IF-THEN-ELSE and DOWHILE/DUNTIL macros, to facilitate employment of proven structured coding habits.
- o Special macros to perform system functions and user desired functions.

These basic features have a significant impact on program production and maintenance costs, lowering them since programs are more readable, transferrable between programmers, debuggable, and maintainable. The transferability alone is vitally important in minimizing dependence on key individuals in periods of illness, other absences, or cessation of employment.

#### 9.4.3 Documentation and Review

Complete system level and subprogram level documentation is required. However, it must be emphasized that in order to meet memory and speed constraints, and to be cost effective, top-down development must be enforced. That is to say, detailed design is considered after higher level functions are designed, documented, and reviewed. Implementation and checkout of higher level functions may be underway concurrent with

lower level design efforts.

Review of design documents is customary and must occur. In addition, experience shows that review of program code itself is also required. A peer-level programmer reviews all code written by another individual, prior to and during debugging, to assist the coder in finding bugs and to assure the code performs exactly as per documented specifications.

#### 9.4.4 Programming Support Center

The Programming Support Center (PSC) must permit macro assembly of micro and minicode, module linkage editing, program loading, and debugging.

The degree to which the center facilitates this depends on schedule constraints and debugging cost constraints. On the other hand, a simpler (cheaper) PSC may cost far less than extra programmer expense and schedule delays.

A minimal PSC is configured in Figure 9-5. The minimal PSC uses a remotely located time-sharing service for cross-assembling and cross-link editing macroassembly code for the microprograms and miniprograms. This center is not self-contained, but has a minimum complement of equipment. Local equipment includes a debugging console (teletype or CRT) for symbolic and hexadecimal breakpointing, patching, memory examination, and tracing of minicode; a medium speed printer for tracing and core dumps; a programmer's panel for hexadecimal only breakpointing, examination, and patching of both micro and mini code; a prototype computer with RAM (alterable memory, micro and mini); and a set of peripheral simulators to perform as the end-item Operator's Panel, sensors and servos.

The minimal PSC required time-shared program update and macro-assembly, cross-link editing, and binary tape generation. The prototype computer's read/write memory is manually loaded with a binary tape for a debugging session. Only a single programmer can debug at a given time, with only one computer available. A typical debugging session will discover five to ten errors before reassembly. Typical segments of code checked per debugging session are 50-100 lines of code. Typical

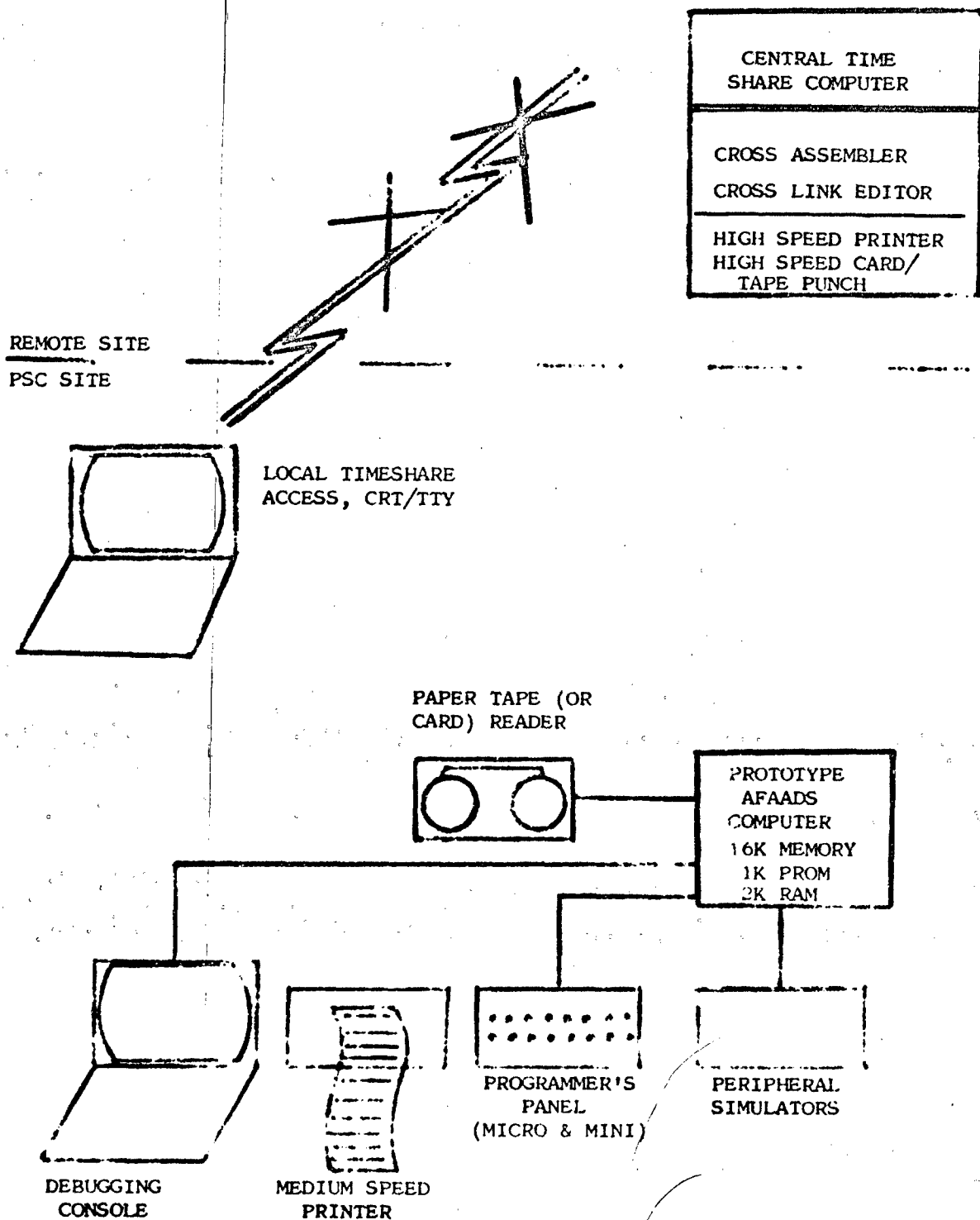


Figure 9-5. Minimal Programming Support Center

turnaround times for a timesharing service to produce a new program tape is several hours, due to the requirement for remote site printing and punching. (Other configurations, such as a local punch, are feasible, but generally slower due to the slow speeds of remote devices.) This system allows a typical programmer to debug 50-100 lines of code per day. Two programmers would collectively be able to debug 100-200 lines of code per day and fully utilize a single prototype computer.

A better PSC configuration is shown in Figure 9-6. This PSC has a movable-arm disc storage subsystem, permitting local macro-assembly, cross-link editing, and debugging in the same computer. With this type of setup, programmer debugging speed is considerably increased due to ease of reassembling after finding a small number of errors. Experience shows that programmers with access to a disc-supported PSC can debug 200-500 lines of code per day.

The disc-based PSC requires only a minimum operating system, to load the macro assembler, link editor, and binary programs, and to manage disc file storage.

Either of the two PSCs may be further supported by a microcode simulator to run on the time sharing system, permitting early checkout of microprograms before complete availability of the PSC equipment.

#### 9.4.5 Field Employment Support

Field employment support requires equipping depot-level centers with adequate test equipment to service the computer itself and with a mechanism to load new versions of program code into the computer. The level of equipment required can only be specified when details of final equipment chores are known.

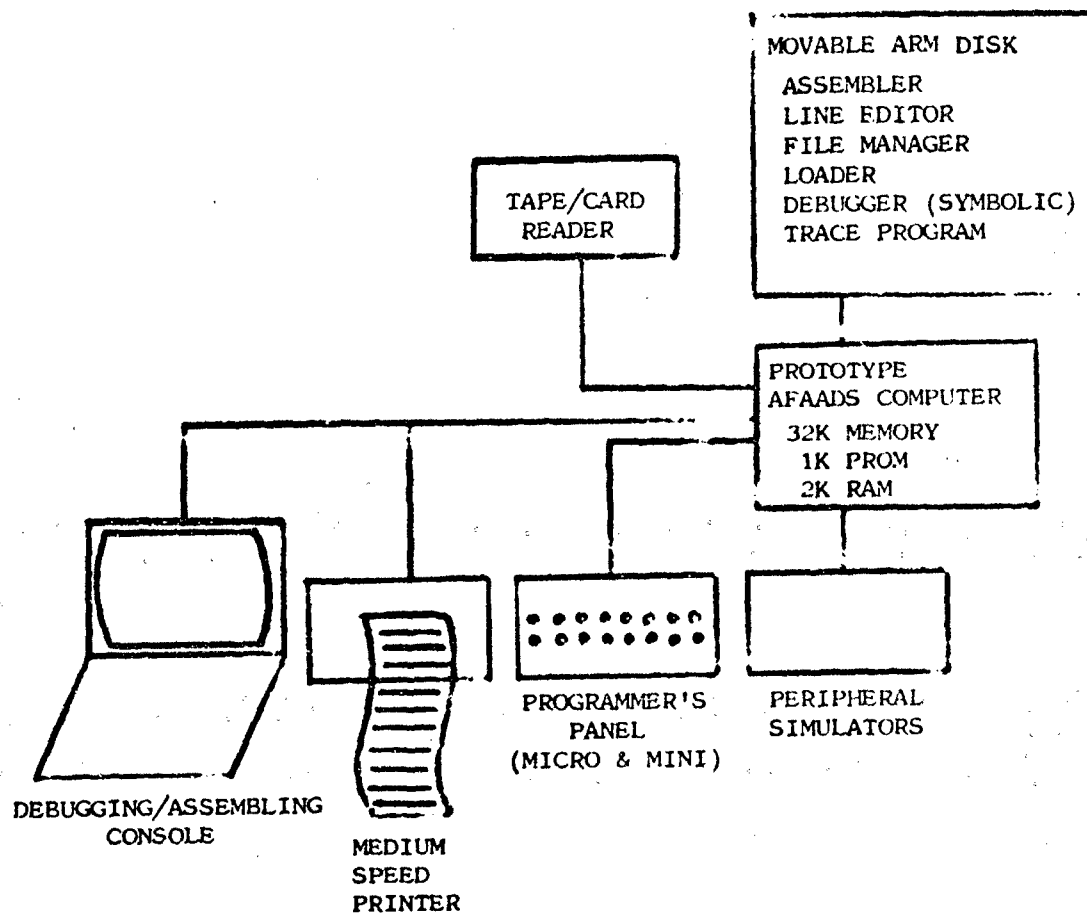


Figure 9-6. Non-Minimal Programming Support Center

## SECTION 10

### HARDWARE - AFAADS COMPUTER

The description of the AFAADS computer hardware concepts begins with an orientation of the computer within AFAADS. It then proceeds into various details. The description is in specific positive terms. This approach permits a positive communication to the designer of the computer and still gives him the latitude for modifications which may be necessary when he selects the circuits, etc.

The topics covered are:

- a) Para 10.1 Key Features of the AFAADS Computer
- b) Para 10.2 AFAADS System Block Diagram and Interfaces
- c) Para 10.3 Computer Requirements and Design Objectives
- d) Para 10.4 Computer Block Diagram and Physical Description
- e) Para 10.5 Computer Operation
- f) Para 10.6 Trade-offs
- g) Para 10.7 Growth Capability
- h) Para 10.8 Next Phase

#### 10.1 KEY FEATURES OF THE AFAADS COMPUTER

The key feature of the AFAADS digital computer are listed below.

<u>Category</u>	<u>Features</u>
Computer Type	Mini-computer, Modular, stored program
Control	Micro program
Logic	MSI (Medium Scale Integration)
Packaging	Self-contained with operator's panel as an integral part of the front. Militarized
Arithmetic	Parallel, binary, fixed point Numbers are signed integers Negative numbers in two's complement 16 process registers used as accum- ulators, overlapped with 7 index registers.

## 10.1 KEY FEATURES OF THE AFAADS COMPUTER

<u>Category</u>	<u>Features</u>
Word Length	Memory word - 16 bits Instruction - 16 and 32 bits Operands - 32, 16, 8 and 1 bits
Program Memory	16,384 words of 16 bits + 2 parity bits
Operand Memory	2,048 words of 16 bits + 2 parity bits
Micro Control Memory	1,024 words of 48 bits
Data Handling	Bit, byte, and word instructions Logical operations Multiple bit shift instructions
Maintenance	Continuous system self-test
Clock	Hardware interval timer Software timers
Input/Output	Digital (A-D converters in the peripheral equipments)
Growth Capability	Modular, expandable memories for the Program, Operand and Micro Control Memories

## 10.2 AFAADS SYSTEM BLOCK DIAGRAM AND INTERFACES

Figure 10-1 shows the computer oriented interconnection block diagram. Note, only the computer interfaces are shown in this diagram. The interfaces to the sensors and servos are assumed to be digitally buffed within the sensors and servos. This interface is of the multiplexed type, parallel bits by device.

## 10.3 COMPUTER REQUIREMENTS AND DESIGN OBJECTIVES

The AFAADS digital computer design concepts have been developed with the following design requirements and objectives in mind.

### 10.3.1 Cost Objective

Small fraction of total AFAADS, probably considerable less than \$35,000 in production quantities.

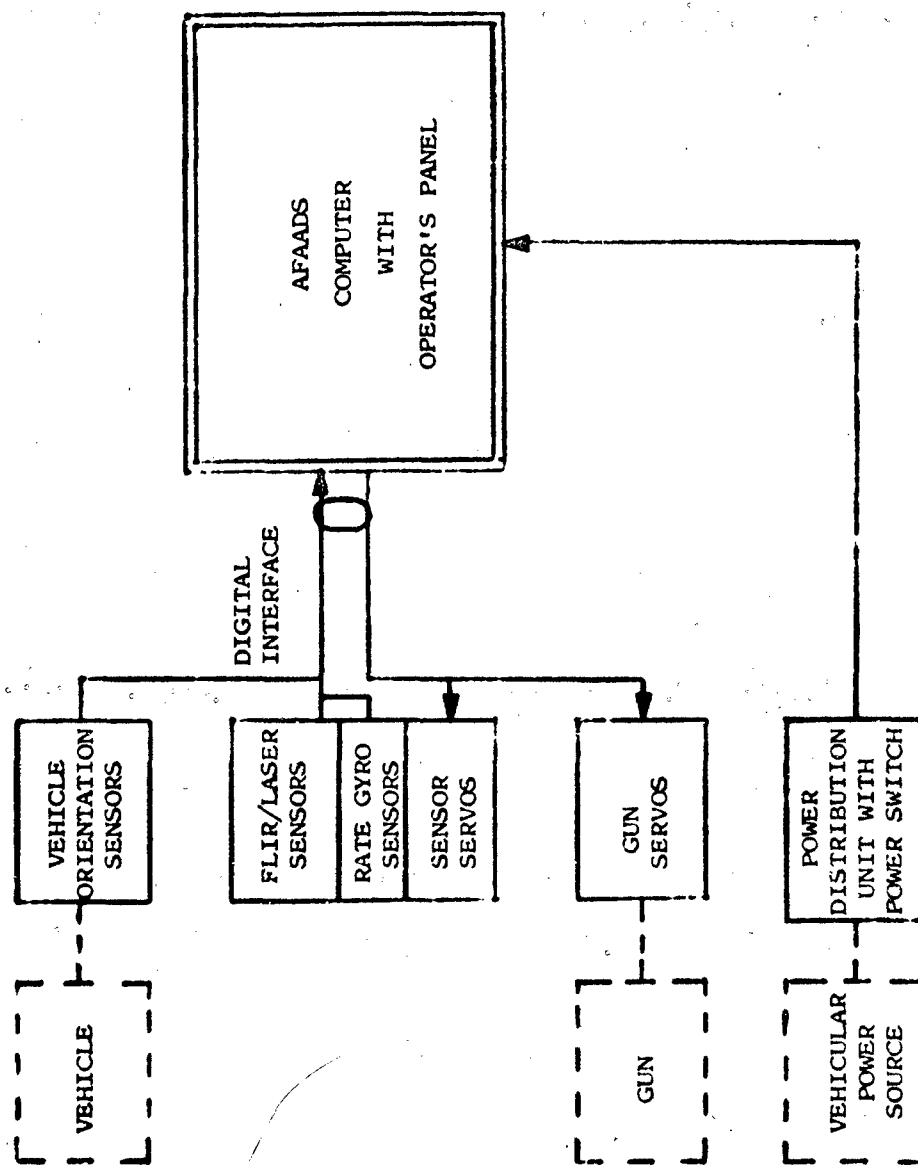


Figure 10-1. AFAADS Computer-Oriented Interconnection Block Diagram

### 10.3.2 Physical Requirements and Objectives

#### A. Units

1. Computer: Single Unit
2. Operator's Panel: Integral with computer, hinged cover for protection and sunshade.

#### B. Size

As small as possible with specific design objectives of:

1. Depth: less than 19 in.
2. Width: less than 10 in.
3. Height: less than 7 in.

#### C. Weight

Less than 50 lbs.

#### D. Power

1. Vehicular Power: 28 volts (19 to 32 volts)
2. Power Switch: On Power Distribution Unit.
3. Over/under voltage sensor for computer shut down:  
Within the computer.
4. Fuses or Circuit Breakers: On Power Distribution Unit.

#### E. Connectors and Cables

1. To Power Distribution Unit: Single connector and single cable.
2. To Sensors and Servos: Single connector and cable to the various interfacing units.
3. To Test Equipment: Single internal connector.
4. To Programmers Console: Single interval connector (part of same connector used with test equipment).

### 10.3.3 Input/Output Requirements

#### A. Type: Digital

#### B. Buffer:

1. Location: In the sensors and the servos, not a part of the computer interface.
2. Sampling: Strobed by computer.
3. Combining: Multiplexed

4. Transfer: Parallel by bit

C. Input: 10 sets of up to 16 bits per set

D. Output: 4 sets of up to 16 bits per set

E. Timing: 1/10 second sampling by computer.

10.3.4 Design Requirements for the Operator's Panel

A. Physical: Integral with front surface of the computer

B. Keyboard/Display Portion

1. Numeric Keys: 10

2. Control Keys: 2

3. Clear/Enter Keys: 2

4. Control Lights: 3

5. Digital Display:

(a) Digits: 8

(b) Segments per Digit: 7 plus point

C. Other Pushbuttons

1. Interpretation: Encoded

2. Sampled: By computer

3. Interlocked: By logic

4. Without Lights: 3

5. With Indicator Lights: 36

D. Other Indicators (in addition to lighted push buttons)

1. No.: 26, all with lights

2. Digit Display

(a) Digits: 4

(b) Segments per Digit: 7 (no point)

3. Source: Computer driven

E. Controls

1. Functions: Brightness Control

2. Power Switch: None, located on Power Distribution Unit

F. Timing

1. Sampling Period: 0.1 sec.
2. Source: Computer driven

G. Overall Panel Lighting

1. Day Use: Ambient
2. Night Use: Provision, details to be worked out.

10.3.5 Processing Objectives and Requirements

A. Input/Output Devices and the Operator's Panel

1. Control: By software
2. Sampling Period: 0.1 second each

B. Interrupts

1. Power restore
2. Power fault
3. Interval timer
4. Not by pushbuttons

C. Computation

1. Control: By software
2. Instruction Time:
  - (a) With indexing: 2 to 5 us (desired)
  - (b) Without indexing: 0.5 us (alternate)

D. Program Memory (Program Storage and Fixed Constants)

1. Type:
  - (a) Development Phase: Programmable Read Only Memory (PROM)
  - (b) Production Phase: PROM or ROM
2. Size: 16 K words of 16 bits
3. Access Time: 1 micro second

E. Operand Memory (Variable Parameters and Work Space Storage)

1. Type: Non volatile, core
2. Size: 2,048 words of 16 bits
3. Access Time: 0.5 microsecond
4. Cycle Time: 1.2 microseconds

F. Micro Control Memory (Micro Programs)

1. Type:

- (a) Development: Programmable Read Only Memory (PROM)
- (b) Production: PROM or ROM

2. Size 1,024 words of 48 bits

3. Access Time: 75 or 60 nano seconds

G. Interval Timer

1. Size: 16 bit register (counter), presettable

2. Update Period: 100 micro second update (decrement)

3. Interrupt: On underflow

4. Maximum Time: About six seconds

5. Additional Timers: By Software

10.3.6 Maintainability and Logistic Support Concepts

A. Maintenance Concept: Depot level (hardware and software), none in field.

B. Spare parts: At depot level

C. Maintenance Philosophy: Unit replacement

D. Operational Test:

1. Automatic: Computer self-test as part of operational software.

2. Operator:

(a) Computer: Operator initiated Confidence Test of computer.

(b) System: Dynamic System Test of complete AFAADS gun system.

10.3.7 Requirements for Training and Manuals

TBD

10.3.8 Environmental Requirements

A. Operating Conditions: Ground mobile/stationary operational

B. Specification: Applicable for forward area equipments

C. Radiation: EMI/TEMPEST

D. Transportable: Air, sea, ground

#### 10.4 COMPUTER BLOCK DIAGRAM AND PHYSICAL DESCRIPTION

This subsection provides a description of the computer from both a block diagram viewpoint and physically.

Figure 10-1, as previously described, shows the AFAADS computer interconnections but without any details relative to the computer itself. Now, in Figure 10-2, a detail block diagram of the computer is shown. The computer is partitioned into the Operator's Panel, the Power Conditioner and eight logic cards. (A total of ten blocks outlined with solid lines).

The block diagram also shows the interface to the Power Distribution Unit, the sensors (input) and the servos (output). The input and output interfaces of the computer are digital. It is assumed that digital buffers are provided within the sensors and servos. Thus, data can be strobed directly in and out of the computer.

Figure 10-3 shows an artist's sketch of the computer mock-up. The computer consists of a housing with a cover for the Operator's Panel, and two connectors. Inside the computer are the logic cards and the power conditioner.

The description (below) of the computer functions uses the block diagram, Figure 10-2, as a basis as well as the physical layout, Figure 10-3.

The computer consists physically of the following parts which will be described in detail in the referenced subsections.

- A. Housing with Cover (Section 10.4.1)
  - 1. Power Connector
  - 2. Input/Output Connector
  - 3. Input Signals
  - 4. Output Signals
- B. Operator's Panel (Section 10.4.2)
- C. Power Conditioner (Section 10.4.3)
- D. 1 Input/Output Logic Card (Section 10.4.4)
- E. 2 PROM Logic Cards (Section 10.4.5)

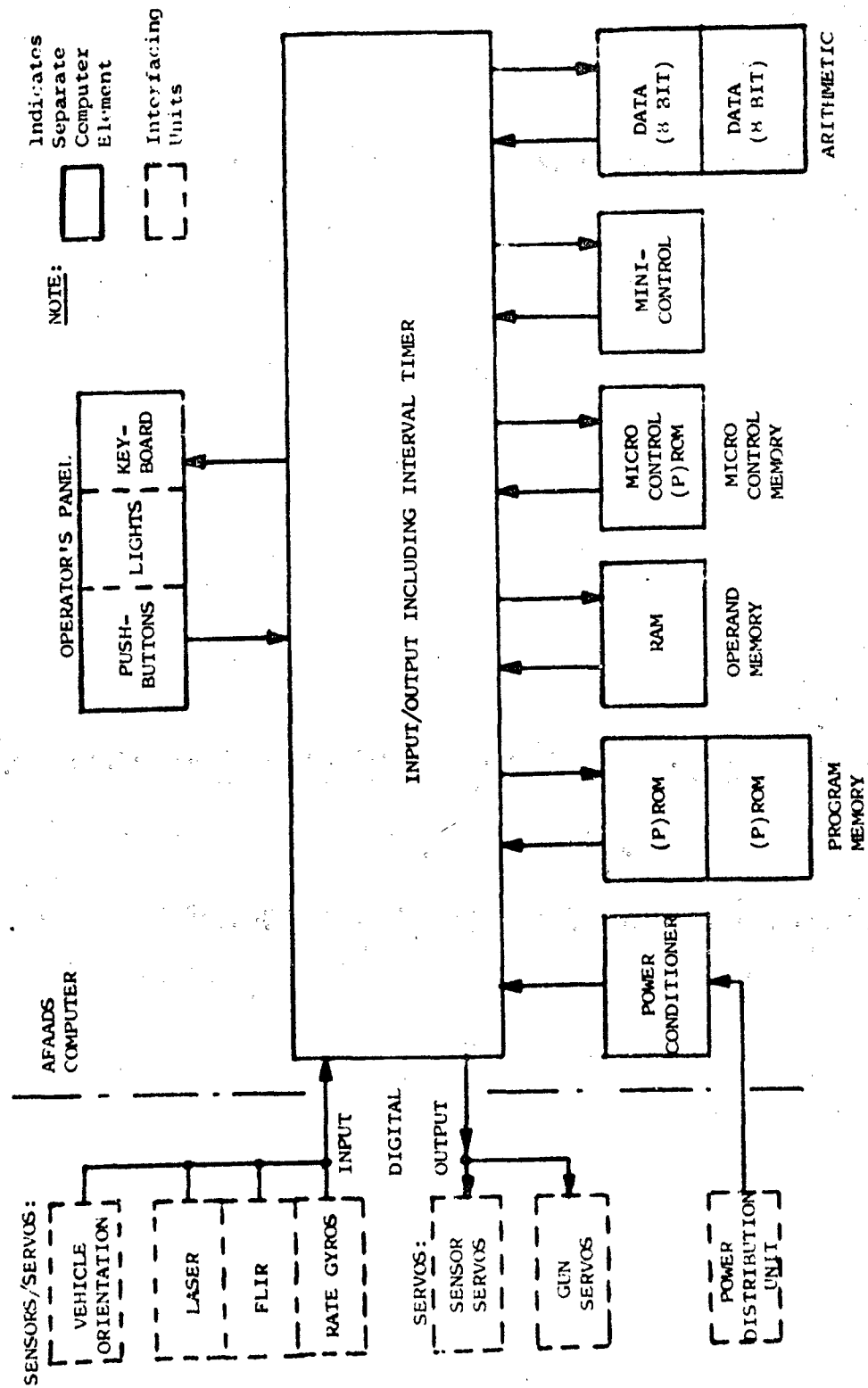


Figure 10-2. AFADS Computer Block Diagram

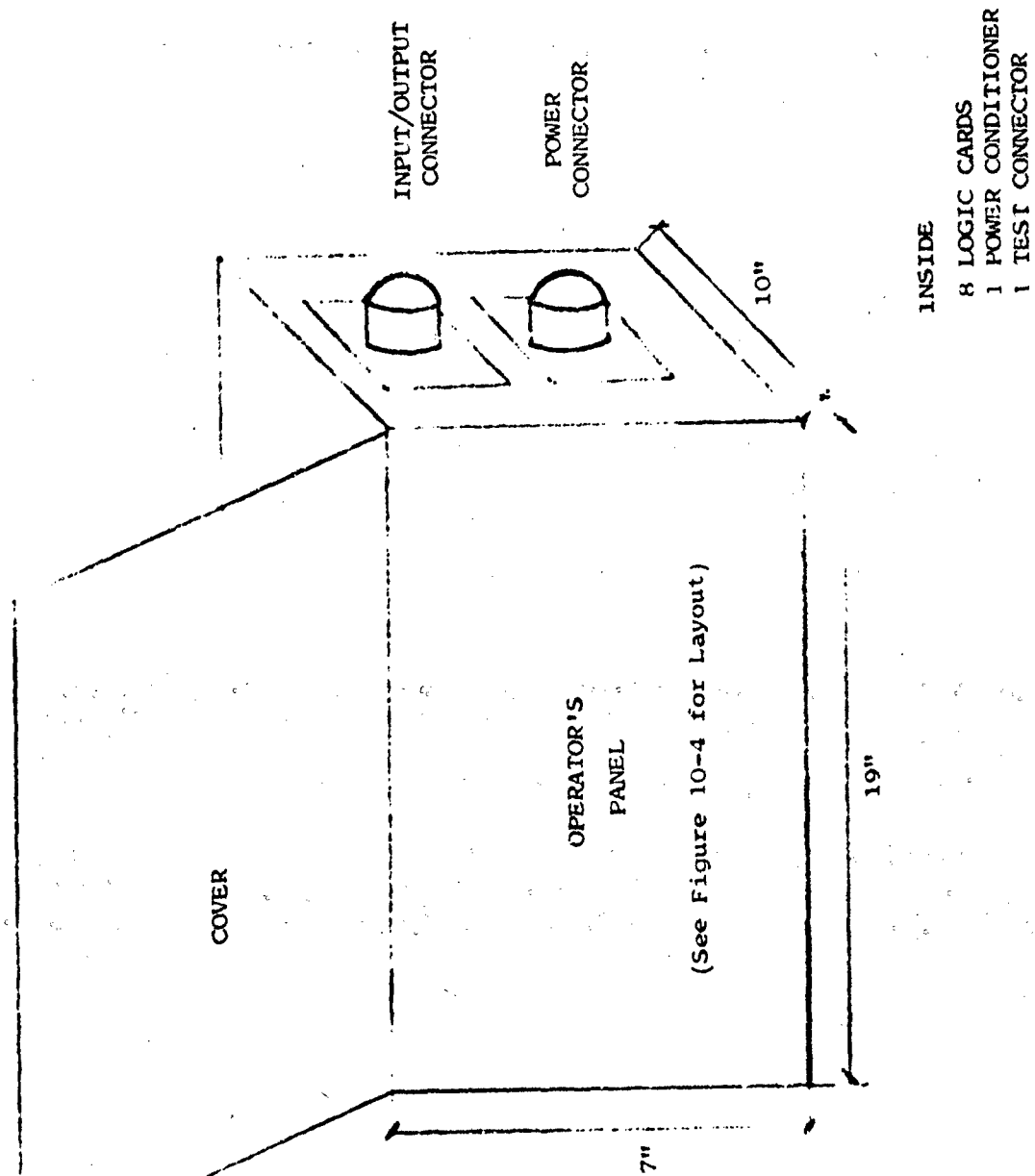


Figure 10-3. AFAADS COMPUTER MOCKUP

- F. 1 RAM Logic Card (Section 10.4.6)
- G. 1 Micro Control Logic Card (Section 10.4.7)
- H. 1 Mini-Control Logic Card (Section 10.4.8)
- I. 2 Data Logic Cards (Section 10.4.9)
- J. Ancillaries
  - a. Test Connector (Inside) (Section 10.4.10)
  - b. Wiring

Each logic card measures approximately 6-1/2 x 8-1/2 inches, mounted on 3/4 inch centers.

The flexible computer architecture provides for growth and expansion by adding the following logic card types: PROM, RAM, Micro Control and Data Cards, as described in Section 10.7.

#### 10.4.1 Housing with Cover

The computer, Figure 10-3, is a self-contained unit. A cover over the Operator's Panel provides protection during shipment and movement in the field. When the cover is opened, it gives access to the Operator's Panel. The cover may be locked in various positions to provide a sun shade. The requirement for illumination of the operator's panel at night will be determined during the development phase.

Appropriate mounting of the computer to the gun will be provided. Details of the mount, including the probable requirement for shock mounts, are deferred to the design phase and are not shown.

The housing has two external connectors. One for power and one for the signals. Each has a screw-on cover for shipment.

The principal elements of this housing are now discussed.

#### A. Power Connector

Prime power for the computer is obtained from the vehicle power source. The power switch and the fuses or circuit breakers are assumed to be a part of the Power Distribution Unit. This location of the power switch, rather than on the Operator's Panel, was chosen to lessen the chance of accidental system shut down. The computer is designed to start to

operate automatically when power is applied and to shut down orderly when power is removed.

B. Input/Output Connector

It is assumed that the input and output signals are all in a single cable. This cable may be built in two variations.

a. Straight Through Cable

For instance the cable could be routed from the computer to the sensor unit. A second connector on the sensor unit would route the output signals to the gun servos. From there they go to the vehicle orientation sensors.

b. Branched Cable

The single cable from the computer is branched. One branch routes to the sensors, the other branches are routed to the gun servos and vehicle orientation sensors.

C. Input Signals

The input signals received by the computer are listed in Table 10-I by sensor. For each sensor, the specific signals are given. Each signal is described by name, by the algebraic symbol used elsewhere in this computer concepts analysis, by peripheral device number used in the discussion of Section 8, and by the number of significant digits including sign bit where applicable. These signals are all sampled by the computer at 0.1 second intervals.

D. Output Signals

The signals outputted by the computer to the servos are listed in Table 10-II. The format of this table is identical to that used to describe the Input Signals in the preceeding paragraph. The output rate is approximately 10 times per second as determined by the computer program, not by the timer.

Table 10-I Computer Input Signals

Sensor	Signal	Periph. Device	Size
Laser	Target Position: Distance $D(j)$	D3.a.2	16 bits
	Miss Distance: Range $\Delta D(mk)$	D3.b.2	13 bits
FLIR	Target Position: Azimuth $A(j)$	D3.a.1	15 bits
	Target Position: Elevation $E(j)$	D3.a.1	14 bits
	Miss Distance: Transverse $\Delta T(mk)$	D3.b.1	10 bits
	Miss Distance: Elevation $\Delta E(mk)$	D3.b.1	9 bits
Gyro	Angular Transverse Rate $\dot{1}(j)$	D4	9 bits
	Angular Elevation Rate $\dot{E}(j)$	D4	9 bits
Vehicle	Cant $\phi(v)$	D7	7 bits
	Pitch $\theta(v)$	D7	7 bits

Table 10-II Computer Output Signals

Servo	Signal	Periph. Device	Size
Gun Servos	Azimuth Lead Angle $\delta(j)$	D5	14 bits
	Elevation Lead Angle $\sigma(j)$	D5	14 bits
Sensor Servos	Azimuth Rate $\dot{A}(j)$	D6	9 bits
	Elevation Rate $\dot{E}(j)$	D6	9 bits

#### 10.4.2 Operator's Panel

Figure 10-4 shows the tentative layout of the Operator's panel. The panel layout has taken into consideration the logic sequencing of pushbuttons and indicators when the operator is using the panel.

The panel consists of the following parts:

- P = Pushbutton without light
- P/L = Pushbutton with light
- I = Indicator (light)
- LED = Light Emitting Diode display
- BRT = Brightness Control

The parts on the operator's panel are grouped into Areas 1 through 6 for reference.

##### A. Initialization (Area 1)

- a. 20 P/L
- b. 8 I
- c. 1 LED of 4 digits

##### B. Alarm and Fire Control Status (Area 2)

- 9 I

##### C. Manual Fire Control (Area 3)

- 1 P/L

##### D. Test (Area 4)

- a. 2 P
- b. 10 P/L
- c. 4 I
- d. 1 BRT

##### E. Semi-Automatic Fire Control (Area 5)

- a. 1 P
- b. 5 P/L
- c. 5 I

##### F. Keyboard (Area 6)

- a. 14 P
- b. 3 I
- c. 1 LED of 8 digits

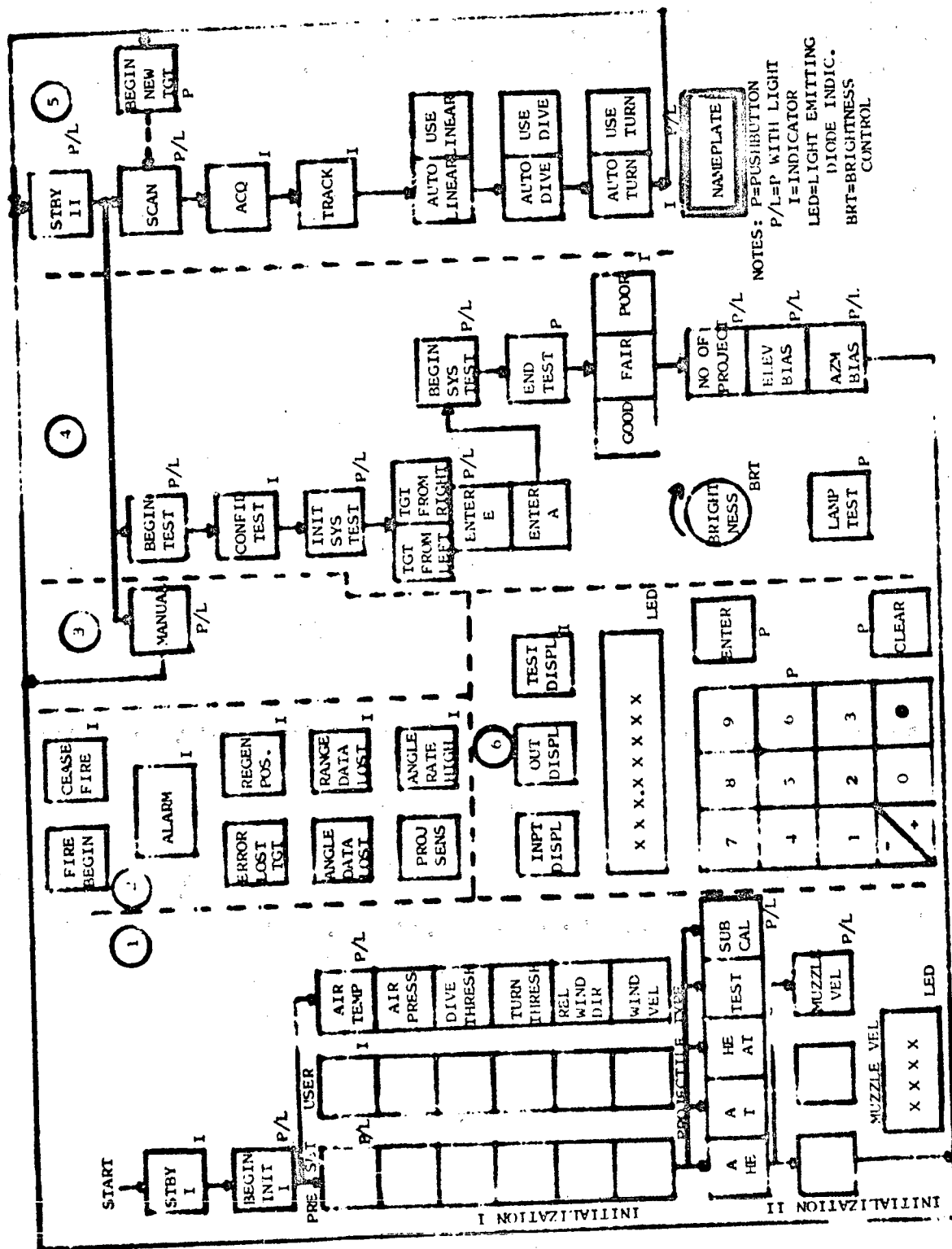


Figure 10-4. Operator's Panel

#### G. Totals

a. Pushbuttons w/o light	17
b. Pushbuttons with light	36
c. Pushbutton codes (total)	53 = 6 bit code
d. Indicators	29
e. Lights in pushbuttons	36
f. Lights (total)	65 = 5 sets of 16
g. LED	2 of 8 and 4 digits
h. Brightness Control	1

#### 10.4.3 Power Conditioner

See Figure 10-2 for reference.

The functions performed by the power conditioner are:

- a. Regulate incoming voltage
- b. Output voltage to logic
- c. Sense over and under voltage
- d. Over voltage protection
- e. Generate start signal or power-up
- f. Generate shutdown signal when voltage is out of tolerance

#### 10.4.4 Input/Output Logic Card

See Figure 10-2 for reference.

This card provides the following functions:

- a. Input from peripheral devices and pushbuttons.
- b. Output to peripheral devices, lights and LED indicators
- c. Communication between other logic cards
- d. Interval Timer

#### 10.4.5 PROM Logic Cards

See Figure 10-2 for reference.

The two PROM Logic Cards are either

ROM (Read Only Memory), or

PROM (Programmable Read Only Memory)

During development PROM is preferred because its contents are

erasable and can be reprogrammed. The ROM may be selected for production.

There are either

2 ROM cards of 8,192 words each, or

4 PROM cards of 4,096 words each

for a total of 16,384 words. Each word is 16 bits. The difference in the number of cards is due to packaging, heat dissipation and the bit density per chip.

The PROM Logic Cards contain:

- a. Programs
- b. Constants (fixed)
- c. Tables (fixed)

#### 10.4.6 RAM Logic Card

See Figure 10-2 for reference.

The RAM Logic Card contains a

RAM (Random Access Memory) of 2,084 words,

16 bits per word associated electronic circuits

The RAM is built as a core memory.

The contents are non-volatile which means its contents are preserved when the power is shot down. Further, the contents are electrically alterable and access to each word in the RAM is by an address. The sequence of addresses can be at random and does not need to be sequential.

The RAM Cards are used for:

- a. Variable parameters (must be preserved during inadvertent shut down)
- b. Work space for calculations

#### 10.4.7 Micro Control Logic Card

See Figure 10-2 for reference.

The Micro Control Logic Card contains either

ROM (Read Only Memory) or

PROM (Programmable Read Only Memory)

The selection criteris is the same as that described for the PROM Logic Card (see above).

The card contains

1024 words, 48 bits per word microinstruction memory

Associated microcontrol logic of:

- a. Microinstruction registers
- b. Four level interrupt structure
- c. 16 address push/pop stack
- d. Shift control

It contains the micro code for controlling the computer's logic which mainly performs the fetching and staticizing of instructions as well as the individual steps in executing an instruction.

#### 10.4.8 Mini-Control Logic Card

See Figure 10-2 for reference.

The Mini-Control Logic Card contains the logic for the orderly sequencing of the control logic. It contains:

- a. Instruction location address register
- b. Instruction decoder
- c. Operand address register
- d. Control for accessing the PROM and RAM Logic Cards

The control card provides for:

- a. Start when power goes on
- b. Shut down when power goes off
- c. Interrupt handling
- d. Instruction fetch
- e. Instruction execution steering to microcode

#### 10.4.9 Data Logic Cards

See Figure 10-2 for reference.

There are two Data Logic Cards. Each is equipped for 8 data bits. Thus, a total of 16 bits can be processed together.

The Data Logic Cards provide for the following functions:

- a. Operand Registers
- b. 16 bit adder
- c. 16 process registers of 16 bits each
- d. Shift capability
- e. Logic operations

The Data Cards perform mainly:

- a. Arithmetic and logic operations
- b. Address indexing
- c. Shifting

The Data Cards are modular to provide for 8 to 32 bits arithmetic, in increments of 8 bits. Thus, it is adaptable to various applications and has growth capability.

#### 10.4.10 Test Connector

A connector is provided inside the computer for the purpose of connecting test equipment and/or a programmer's console to the computer. This connection is only made at depot level when trouble shooting of the computer becomes necessary.

The test connector provides access to the internal logic of the computer. The test equipment or the programmer's console permits the display of registers and provides control signals for stepping through the individual instructions of the program. The memories are accessible and its contents can be displayed. Thus, both software and hardware errors can be detected and diagnosed.

### 10.5 COMPUTER OPERATION

The operation of the computer hardware is now presented. It complements the material on the Operator's Controls and Displays of Section 7 and on the Software of Section 8. The hardware discussion uses the various elements of the computer shown in the block diagram of figure 10-2 and the Operator's Panel of Figure 10-4. The description is not by card and not by operational function, but rather by processing function or operation. The sequence used is arbitrary and has not been related to the importance of the described operations.

#### 10.5.1 Operator's Panel

##### A. General

It is assumed that only one pushbutton is depressed at a time. The rate of depression is assumed to be less than 5 times a second. Each pushbutton is encoded into a register. A second code can only be entered after the computer has cleared the register. This interlock eliminates erroneous codes. Thus, if a second pushbutton is depressed too fast, it

is ignored; the Operator recognizes this by inappropriate responses. He then depresses the second pushbutton again..

Some pushbuttons, such as on the keyboard, are interlocked by the software. This means that depressing of a key on the keyboard will only be effective when appropriate control buttons have been previously depressed. Also the depressing of many of the pushbuttons in an arbitrary sequence is interlocked by software. All indicator lights, lights in the pushbuttons and LEDs are computer driven.

#### B. Pushbuttons

Programmed instructions read pushbutton switch actions under the control of the interval timer.

It is assumed that only one switch at a time is depressed. Appropriate interlock is provided (see A above). All switches are encoded into a register. The zero code is not used. The code is aligned with the least significant bit in the 16 bit register (only 6 bits are used). The most significant (additional) bit in the register is used for control. Depression of a pushbutton not only enters the code in the register but also resets the most significant (control) bit to "zero."

The computer reads the code and interprets it. A "one" control bit indicates no entry. A "zero" control bit requires action. The computer also clears the register and sets the most significant bit to a "one."

Naturally, this register will be initialized when the power is switched on.

#### C. Keyboard/Display

The keyboard and its display operate under computer control (see also A and B above). After selecting the appropriate control pushbutton, which removes the software interlock (Sec 7, Operator Controls), the operator can employ the keyboard as follows.

The operator depresses the CLEAR button which is recognized by the computer (Figure 10-4). Software extinguishes the TEST DISPL, OUT DISPL, or INPT DISPL indicators and clears the LED display.

Depression of a numeric key causes (by software) display of the number in the rightmost position on the LED display. Each subsequent

key action causes the display to appear shifted left by one digit and the entrance of the new digit in the rightmost position. The program prevents the operator from entering more than 7 digits. Leading zero digits are displayed when the operator depresses the ZERO key.

The POINT key causes the display of a decimal point to the right of the rightmost digit; no shifting takes place. Each subsequent numeric key depression causes shifting of the display including the decimal point.

The depression of the "-/+" key causes a change of sign. A blank digit is assumed to represent a positive sign, a minus sign is displayed for a negative number. Depression of the "-/+" key causes the program to place a minus sign to the left of the most significant, previously entered digit. Repeated depression cause alternately the removal and entering of the minus sign and no shifting of the display. Entering of new numeric digits causes shifting of the display, including the minus (or blank) sign digit.

If the operator makes a mistake, he can clear the display and re-enter the correct number by use of the CLEAR button.

After the operator has composed a good number on the display, he can enter the number into the computer by depressing the ENTER key. The program reads the previously composed number in conjunction with the previously depressed control button. The program clears the display and re-displays the number for verification by the operator. At the same time the INPT DISPL (Input Display) indicator is lit.

The program can also output numbers as the result of depressing pushbuttons. In this case, the OUT DISPL (Output Display) indicator is lit. A limited set of alphabetic symbols can also be outputted as the result of depressing one of the test result pushbuttons. The result is presented on the LED display and the TEST DISPL (Test Display) indicator is lit.

If any one of the above three DISPL (Display) indicators is lit, then the keyboard is inoperative except for CLEAR. Depression of any key does not affect the display. The Operator must first depress the CLEAR key before he is able to prepare a new number for entry.

Each LED digit consists of 7 segments and a point. Thus, a digit consists of 8 bits. Each bit corresponds to a segment or the point. The program is capable of illuminating selectively each segment. This scheme provides for not only displaying numeric symbols but also a limited repertoire of alphabetic symbols. (A four bit "code" would limit the display to 15 digits). The 8 digit display corresponds to 4 words of 16 bits each.

#### D. Indicators

Programmed instructions set and reset flip-flops which drive the indicator lights.

#### E. Alarm

The ALARM is a blinking light indicating that the computer is inoperable. This is self-determined as follows.

The computer program, using an instruction, must deactivate or reset the alarm circuit periodically. The interval between resets should not exceed one-half second. The alarm reset inhibits the blinking of the light. When the program fails to operate correctly it will not reset the alarm. Thus, the light starts to blink.

#### F. Lamp Test

The LAMP TEST pushbutton, when depressed, turns all Operator's Panel lights on. The operator then verifies visually that all lights operate. This pushbutton also intercepts the reset signal to the alarm. Thus, the alarm indicator should blink for visual inspection.

#### G. Brightness Control

This control provides for gradual (or step incremented) adjustment of the brightness of the lights on the indicators and pushbuttons. The adjustment is provided for day or night operations.

#### 10.5.2 Program

A program consists of the set of instructions which control the operation of the computer. The instructions listed below are by classes.

- a. Load and store
- b. Arithmetic, including multiply and divide
- c. Compare and test

- d. Logic
- e. Shift
- f. Branch (transfer)
- g. Input/Output

The instruction format is shown in Figure 10-5. Three of the four formats are 16 bits long. The third or Address Format is 32 bits long. All formats contain an operation code of 8 bits which determines the operation to be performed. The operation code includes fields, which are not shown, which determine the applicable address mode.

Each of the four formats is described in further detail.

Register Format: It is used for register to register operations. The Registers A and B refer to the process registers. For input/output operations Register A is interpreted as a peripheral device address and Register B contains the input/output data.

Literal Format: This format contains a 4 bit literal operand which is applied to the Process Register B.

Address Format: Register B refers to a process register which contains an index. The index is applied to the operand address to obtain a final address. This address is used to access an operand. The operation code then specifies the operation to be performed between the operand and the process register specified by Register A. The operand address can be interpreted as a literal, depending upon the operation code.

Displacement Format: This format provides for the following major operations. Displacement (a) is a transfer, (b) is a branch address; or (c) contains the shift control.

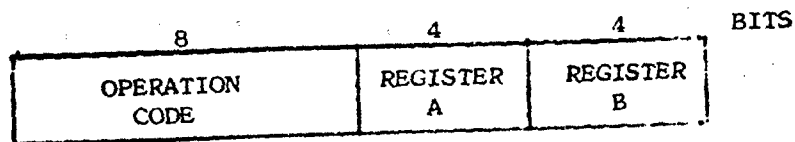
#### 10.5.3 Micro Control

Micro control performs the individual steps as necessary to staticize and execute an instruction. Since the individual steps are micro programmable, it is possible to replace a subroutine (program) by a special instruction which then performs all the necessary steps directly from the micro control program memory.

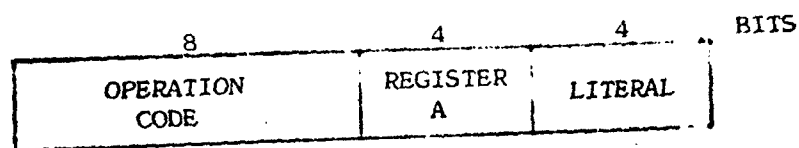
#### 10.5.4 Interrupts

Interrupts permit the suspension of the normal program sequence to perform special functions related to the type of interrupt present.

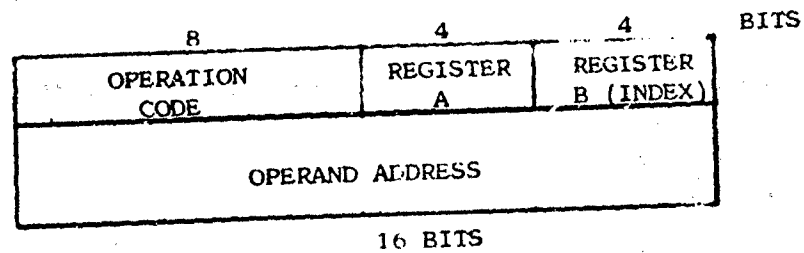
A. REGISTER FORMAT



B. LITERAL FORMAT



C. ADDRESS FORMAT



D. DISPLACEMENT FORMAT

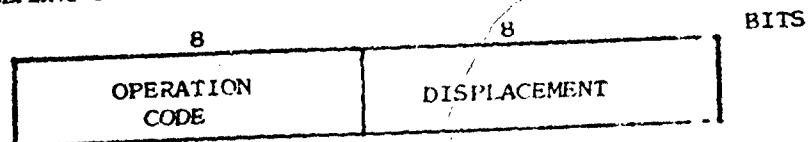


Figure 10-5. Instruction Formats

Interrupt processing depends on a priority network. The assigned priorities include:

- . Power Restore
- . Power Fault
- . Interval Timer

The action of an interrupt is to terminate the current program sequence at the end of the instruction execution cycle, to save pertinent data to allow resumption of normal program operation from the point of interruption, and to transfer control to an instruction located at an address corresponding to the interrupt.

The power restore program starts at location zero which contains a branch and link instruction referring to the initialization routine. This routine includes the initialization of the displays and the indicator lights. It also clears the code in the pushbutton register and sets the interval timer.

The power fault program is initiated when the computer senses over or under voltages and provides for orderly shut down. The power down program provides for the saving of the process registers and the volatile registers.

#### 10.5.5 Interval Timer

A 16 bit register (counter) is preset to any number between 1 and 65,535 by an instruction. The counter is then decremented each 100 micro seconds. When the counter underflows, i.e., goes from zero to negative (the maximum value), then the computer is interrupted. The program takes action such as input data from the peripheral devices and pushbuttons, and again presets the interval timer/register. For instance if the counter is set to 1000, then an interrupt will take place after one hundred milliseconds which corresponds to the sampling period of 0.1 second used elsewhere in this analysis.

The interrupt program may maintain additional software timers as needed which are slaved to the hardware interval timer just discussed.

#### 10.5.6 Input/Output

Inputs from the peripheral devices are under program control. The rate of the input sampling is determined by setting at the interval timer which performs an interrupt (see Subsection 10.5.5).

Programmed instructions read inputs from the sensors into a process register for further processing.

Outputs to the peripheral devices are under program control. The rate of the output is determined by the computations.

Programmed instructions write outputs from a process register to the servos.

Each peripheral device is accessed by an individual address code. The input/output data transfer is assumed to be up to 16 bits in parallel for a given peripheral device address. An associated signal to the peripheral device provides for strobing of the data to and from the computer. During the presence of the strobe signal, it is assumed that the input data does not change within the peripheral device.

Each 16 bit data transfer uses a programmed instruction.

#### 10.5.7 Hardware Tests

The computer hardware is being designed to include three different types of testing:

- a. Self-testing performed automatically in the field under software control
- b. Confidence testing performed in the field as a result of operator initiation and under software control
- c. Depot level maintenance testing performed at depots under the control of maintenance personnel

As indicated the first two are performed in the field and use built-in software fault detection logic. The third requires external test equipment.

Confidence tests are initiated by the operator to verify certain computer functions and its operation as determined by the selected test. Section 13 describes the operator functions associated with these tests.

Self-test is performed automatically under program control on a periodic basis as determined by a software timer. The software timer is slaved to the interval timer. The self-test program performs a test of all major functions and operations in the computer.

A sample list of specific tests that will be carried out in one or more of the three categories of self-test, confidence test, and maintenance test are discussed below. In each case the applicable categories are indicated.

A. Parity Test

All memories have a parity bit associated with each 8 bits. The read-only memories (PROM or ROM) have a predetermined parity, the core memory requires the generation of a parity bit when an operand is stored. Access of all memories requires checking of parity. Wrong parity causes the program to be interrupted and transfer of control to the error program takes place. This test capability lies in all three categories.

B. PROM (or ROM) Test

The contents of the programmable read only memory (PROM) and read only memory (ROM) are not altered during operations. Thus, during initial loading at the factory (or depot), a check sum is calculated and stored in the last location of each memory circuit. When the program performs the PROM (or ROM) Test, it will calculate the check sum and compare it against the prestored check sum. If the two numbers compare, then the contents of the memory is still intact.

Check sums can also be provided for individual parts of memory. A bad check sum is then an indication of which memory part failed. This test is likewise a part of self-testing, confidence testing and maintenance testing.

C. RAM Test

Since the contents of the random access memory (RAM) are alterable during operations, a more elaborate test is required. Let us assume that a certain number of memory words are not used. The test procedure is as follows:

1. Move (first) N words of RAM to the spare area for safe keeping during the test.
2. Clear the N words.
3. Insert the addresses into the addressed words that have been cleared (N addresses into N words).

4. Read the N words and compare with the addresses.
  5. If they do not compare, then there is an error.
  6. If they do compare insert complement addresses in the same N words.
  7. Read the N words and compare with the complemented addresses.
  8. If they do not compare, then there is an error.
  9. If they do compare, move spare area to first N words by restoring (first) N words
  10. Repeat for next N words.
- This test is a part of self testing and confidence testing.

D. Arithmetic Test

This program performs predetermined operations such as add, subtract, compare, shift and compares with prestored results. Both self-testing and confidence testing use this test

E. LED Display Test

The LED Display Test is a part of confidence testing and maintenance testing, but not self-testing. Under current plans it is an integral part of the test, but it could be a separate operator initiated test. The test results can be displayed as all one's on the LED display which means the test is good. Any zero displayed can indicate a bad test result. The position of a zero can indicate to the maintenance man where the trouble is to be expected.

The maintenance man at depot level can perform an additional test by inserting a test number through the keyboard. The result of the test can be displayed on the LED display. Details depend upon the final design of the computer.

F. Alarm

The operational program when detecting a fault or incorrect operation of the computer has the capability to inhibit the resetting of the alarm. The alarm light starts to blink. This is a computer self-test feature only.

#### G. Input/Output Test

There are several techniques for performing routine input/output testing. For instance, the input/output can be tested by providing a "dummy" peripheral device with the sensors and servos. The computer outputs data to the dummy device and then inputs it for comparison.

An alternate approach would use a selective test signal to each peripheral device. Each test signal would be followed by a regularly programmed input instruction to the same peripheral device. The peripheral device would supply a predetermined test pattern (response) for this sequence of operations. These responses would be analyzed for accuracy. Following completion of the test, the software program would again input the operational data.

This test capability is in all three categories.

#### H. Maintenance Testing

In addition to the above described portions of maintenance testing, access to the computer logic for more extensive testing is provided through a test connector inside the computer. The maintenance man, at depot level, is provided with the capability to access registers, display registers, and modify registers by using special test equipment. The test details and the test equipment requirements will be determined during the design phase of AFAADS.

This same access can also be used to verify the computer program through use of a programmer's console tied to the computer via the test connector.

### 10.6 TRADE-OFFS

Various trade-offs were made before selecting the proposed computer design. Many of these are discussed below. For each trade-off discussed below, the individual design choices are first presented. In

every case the selected choice is the first one on the list. Following the list, the major arguments used in the selection process are given.

#### 10.6.1 Program Storage

##### A. Design Choices

1. Programmable Read Only Memory (PROM) - Selected for development. May also be used in production.
2. Read Only Memory (ROM) (for production only)
3. Alterable Storage
4. Pluggable Program
5. Wired-in Program

##### B. Selection Criteria

Choice No. 5 requires rewiring of the unit for each change, also it is voluminous.

No. 4 has mechanical problems and is bulky.

The third choice is desirable but requires a loading unit in the field.

The selected design (Item No. 1) provides for compact program storage with the provision of making program changes during the design phase. For production, the ROM, Item No. 2, may be selected after an appropriate cost trade-off has been performed.

#### 10.6.2 Operand Storage

##### A. Design Choices

1. Core, non volatile - Selected
2. Core and semiconductor
3. Semiconductor

##### B. Selection Criteria

There is a need of alterable storage (RAM - Random Access Memory) for (a) work space and (b) variable parameters. Note, fixed parameters, constants, tables, etc., are colocated with

the program in the Program Storage.

The work space can be volatile but the variable parameters must be stored in a non-volatile media. Presently, only core provides for non-volatile storage.

Semiconductors can be made pseudo non-volatile but require batteries, charger, voltage detectors, and refresh circuitry.

MOS (metal oxide semiconductor) storage has the same speed as core. Bi-polar semiconductor storage is faster than core but requires more power and needs more space than MOS.

One could use two types of storage, but the semiconductor storage plus battery, etc., is larger than the equivalent core for this particular application. Also, fewer different parts (only one type of storage, rather than two) are desirable from a spare parts and maintenance viewpoint.

#### 10.6.3 Minicomputer Control

##### A. Design Choices

1. Minicomputer with instructions and microprogram memory - Selected
2. Microprogram only
3. Wired-in instructions

##### B. Selection Criteria

The third choice is used only in larger computers and requires more logic than a microprogram stored in ROM (read only memory).

No. 2, microprogram only, is wasteful when a program is large because micro code is longer (more bits per word) than an instruction word.

The selected design has the instructions stored in PROM (programmable read only memory), 16 bits per word and the microprograms are stored in ROM or PROM, 48 bits per word.

#### 10.6.4 Push Buttons

##### A. Design Choices

1. Momentary contact - Selected
2. Two positions

##### B. Selection Criteria

Only one type of push button was selected to reduce spare parts. A two position push button, the second choice, is more complex than the selected push button and therefore is more prone to mechanical problems. Also, use of a two position push button requires indicators on each push button to indicate positively the two positions.

#### 10.6.5 Multiposition Switch - Few Positions

(Example: Projectile type selection, one of five choices)

##### A. Design Choices

1. Individual push buttons, with indicators and software interlock - Selected
2. Rotary switch, multiposition

##### B. Selection Criteria

A multipurpose rotary switch handle may slip on its axle which results in misalignment with the labels. It also introduces a different component into inventory. The selected use of push buttons standardizes on the same parts used for other purposes (see Subsection 10.6.4).

In operation, the selection of a particular projectile type which push buttons causes an interrupt to the computer. Software recognizes the new input. On the other hand, use of a rotary switch requires the program to read the switch position at predetermined intervals to see if the operator has turned the switch.

#### 10.6.6 Multipurpose Switch - Many Positions

(Example: Muzzle velocity, 64 choices)

##### A. Design Choices

1. LED (light emitting diode) display with keyboard -  
Selected
2. Two rotary switches
3. Two thumbwheel switches

##### B. Selection Criteria

The selected design uses the keyboard for entry of the number. The number "is" the velocity; and is always displayed. No encoding by the operator or software decoding is required.

The two rejected designs result in complicated labeling such as unit digit is always zero and thousand digit is always one; otherwise, additional switches needed. Also, these switches would be non-standard additional parts.

#### 10.6.7 System Test Results

##### A. Design Choices

1. LED display-Selected
2. Individual indicators with labels

##### B. Selection Criteria

The rejected choice uses more components than the selected one and once selected gives only the fixed, predetermined variations of the test results.

On the other hand the selected LED display design gives full flexibility of display, only limited by programming. It is a standard part, which is an integral part of the keyboard. It requires less panel space and permits meaningful messages to be displayed.

#### 10.6.8 LED Display Code

##### A. Design Choices

1. 8 bit code - Selected
2. 4 bit code per digit plus point position code

##### B. Selection Criteria

The second choice uses a more efficient code but allows for only 15 symbols to be displayed.

The 8-bit code is selected because it permits the display of alphabetic symbols in addition to the numeric symbols. Thus, test results may be displayed using a limited set of alphanumeric symbols. Additionally, the decimal point is associated with each digit and the decoder is eliminated.

#### 10.6.9 LED Illumination

##### A. Design Choices

1. Continuous illumination - Selected
2. Refresh by circulation

##### B. Selection Criteria

The refresh technique is normally used in hand held calculators which operate from batteries. The refreshing is performed by illuminating one digit at a time, which conserves energy.

The selected choice of continuous illumination assumes continuous power is available without restriction. The display is continuously illuminated. It uses a register which contains a bit for each segment of each digit in the display. This amount of hardware is competitive with the hardware needed for the refresh technique. In the rejected system, normally, a shift register with associated counters and decoders perform the refresh. The shifting also requires that the output to the display contains not only the code of

the digit to be displayed, but also a code which specifies the place of the digit. Thus, a comparator detects when the output digit is to be inserted into the shift register. Additionally, a holding register must be provided to hold the output digit until comparison occurs.

#### 10.6.10 Indicators

##### A. Design Choices

1. Two lights - Selected
2. One light

##### B. Selection Criteria

Some parameters such as air temperature may be selected as "Preset" or "User" (entered) by the operator.

The use of two lights, the selected system, gives a positive indication which choice has been selected. On the other hand the rejected system of one light would use a dark indicator as "User" and a lighted indicator as "Preset".

This means the operator must remember and interpret the single indicator into the two options. Thus, the selected technique is much simpler to use at the cost of the additional hardware.

#### 10.6.11 Alarm

##### A. Design Choices

1. Blinking red light - Selected
2. Buzzer
3. All indicators blinking
4. Red light, steady

##### B. Selection Criteria

No. 4 on the list is not enough of an attention getter.

No. 3 requires the program to be working.

No. 2 is bad for field use because of noisy environment and

is hard to seal in an all-weather environment.

The selected choice uses the program to stimulate this monitor. When the program fails to stimulate it, it will blink. The LAMP TEST push button, when depressed, intercepts the monitoring signal and causes the lamp to blink. This shows the Operator that the lamp is operable.

#### 10.6.12 Input/Output Interface

##### A. Design Choices

1. Parallel interface - Selected
2. Serial interface

##### B. Selection Criteria

A parallel input/output interface has been selected because it requires simpler control logic sequencing as compared to the rejected serial interface. However, the disadvantage of a parallel interface is in the 16 data lines required in the input/output cable. However, the input/output cable is of short length in AFAADS.

A serial interface uses only one data line in the input/output cable and associated control lines. The data transfer is serial by bit. This requires synchronization of shift registers and requires more time to transfer the data to and from a peripheral device as compared with a parallel interface.

This trade-off should be re-evaluated during the detail design phase. The final selection of the interface does not affect the overall operation of the computer.

#### 10.7 GROWTH CAPABILITY

The computer architecture (see Figure 10-2) is built from modular blocks. These blocks are expandable and provide for growth. These growth capabilities are explained below:

- A. Data Logic Card: Each Data Logic Card provides for an additional 8 bits in the processing words. Thus the computer can be expanded in a direct manner to perform operations on 24 or 32 bit words.
- B. PROM and RAM Cards: Additional cards will expand either type of memory by 4,096 or 8,192 words. The maximum size without a paging card (see below) is 32,768 words. Larger sizes are accommodated by paging.
- C. Paging Cards: This card provides for expansion of memory beyond 32,768 words.
- D. Microcontrol Card: This card provides for additional PROM which may contain additional micro code.

#### 10.8 NEXT PHASE

It is recommended that the next phase of this project should include the following tasks:

##### 10.8.1 Mechanical

Detail mechanical construction of the computer would include:

- A. Size of computer
- B. Operator's panel layout with emphasis on human factors
- C. Cover design
- D. Mounting of computer on the gun system

##### 10.8.2 Electrical and Circuits

- A. Selection of circuits
- B. Selection of card connectors
- C. Selection of wiring
- D. Power conditioner design

##### 10.8.3 Architecture

- A. Finalize the instruction set
- B. Instruction format detail
- C. Complete functional description

10.8.4 Logic Design

- A. Detail block diagram
- B. Logic flow diagrams
- C. Logic equations
- D. Wire list

10.8.5 Parts Count/Cost Estimate

The description of the computer given in this report is sufficient to provide an estimate of parts count as well as the basis for a cost estimate to build and manufacture the computer.

10.8.6 Build Prototype

Procure parts and fabricate.

10.8.7 Test

Conduct prototype testing, both hardware and computer programs (software).

## SECTION 11

### SEMI-AUTOMATIC FIRE CONTROL ALGORITHMS

The real time processing requirements for the software and hardware computer design concepts, as discussed in the previous two sections, are based upon a set of algorithms required to provide semi-automatic anti-aircraft fire. This section discusses these algorithms in detail, both as separate processing entities, and, more importantly, as an integrated system.

#### 11.1 ALGORITHM SELECTION

In order to develop the software and hardware computer concepts, a specific set of algorithms had to be chosen. This was done, knowing full well that an optimum choice was probably not made in every case. Several areas, in fact, are known to be prime candidates for further exploration and analysis (See recommendations in Section 2.2). The software and hardware design concepts are sufficiently modular and fluid at this time so that changes can be readily incorporated.

The primary selection basis for the semi-automatic fire control algorithms was the analytic results developed during the AFAADS contract and reported in Volume I of this report and in the analysis volumes of each of the two previous reports (Ref. 1 and 2). Of prime importance are the development of (a) the Vista algorithm for measuring the projectile miss distance due to bias errors alone (Section 3 of Volume I and Section 6 of Ref. 2) and (b) the bias correction algorithm using Kalman prediction for developing the exact azimuth, elevation, and muzzle velocity correction factors to the fire control ballistic solution (Volume I, Section 3). Other examples used in the following fire control algorithms are target prediction based upon energy conservation during a diving attack and constant turn prediction. (The energy conservation analysis was verified in the FACT tests at Naval Weapons Center, China Lake. - See Section 4 of Volume I).

The ballistic solution for the 35mm Oerlikon gun in terms of a fifth order polynomial was developed as well as prediction thresholds, wind corrections, gravity drop corrections, etc.

For those few cases where complex or non-obvious algorithms are used, their development is presented in appendices to this volume.

## 11.2 PROCESSING FUNCTIONS

The following is a list of the principal processing functions and logic decisions that have been incorporated into the AFAADS Semi-Automatic Fire Control Mode. Also included are several areas where further analysis is warranted.

- a. Target Tracking and Track Smoothing - a fixed length polynomial solution. (Adaptive Kalman filtering and other processes should also be investigated as alternates.)
- b. Track Prediction
  - (i) Linear
  - (ii) Dive or energy conservation
  - (iii) Constant turn
  - (iv) Defense of a known point (growth feature)
- c. Ballistic Solution - Fifth order polynomial matching 35mm Oerlikon ballistic tables (other approximations should be investigated).
- d. Gravity Drop correction to the ballistic solution
- e. Down and Cross Wind correction to the ballistic solution.
- f. Vehicle Pitch and Cant corrections to the ballistic solution.
- g. Coordinate Conversions - (i) polar to Cartesian and back again.
- h. Regenerative Target Position and Velocity.
- i. Sensing Missed Data Points - target behind obstacle and/or intermittent sensor operation (data goodness criteria should also be considered).
- j. Projectile Miss Distance Computation (Vista algorithm).
- k. Bias Correction Algorithm to give bias corrections in azimuth, elevation, and muzzle velocity.
- l. Logic determining validity of "goodness" of a projectile miss distance measurement for bias correction processing. (The validity criteria needs further analysis.)
- m. Air temperature and pressure corrections to the ballistic solution.
- n. Muzzle velocity correction to the ballistic solution.

### 11.3 THE REAL TIME FIRE CONTROL ALGORITHMS

A simplified block diagram of the AFAADS real time fire control signal flow is given in Figure 11-1. This figure and the detailed block diagram of Figure 11-2 (with the operator interface functions detailed in Figure 11-3) present the functional or signal flow. They do not present the software logic flow. There is no indication on these figures of the time relationship between parallel processes such as is required in the software logic. However, the software must perform all functions indicated between input data sample periods (every 0.1 second).

Note that the two drawings use approximately the same layout arrangements. Correlation between the simplified blocks of Figure 11-1, indicated by the block letter (A, B, C, ...), and the detailed blocks of Figure 11-2 and Figure 11-3, indicated by Roman Numerals (I, II, III, ...), is thus simplified. In addition, Table 11-1 provides an exact correlation between the blocks of the three figures.

In the following pages, the algorithms required for each of the detailed blocks of Figure 11-2 (and 11-3) are presented in numeric order. The same format of presentation, covering each of the 8 items listed, is used throughout:

1. Function: A brief description of the function performed by that block.
2. Frequency: How often the function must be executed.
3. Inputs: A list of the input signals to the block, where they came from (another block or a peripheral device), signal magnitude, and signal accuracy (if known).
4. Constants: A list of the fixed and temporary constants used by the algorithms of the block. Magnitudes are also given.
5. Operations Performed: The actual algorithms performed. Computer logic flows may be included but, if included, they apply to the block under discussion and not to the overall fire control problem.
6. Outputs: Same format as Inputs.
7. Comments: Any remarks dealing with the processing in block.
8. References: Source material for the algorithm may be included.

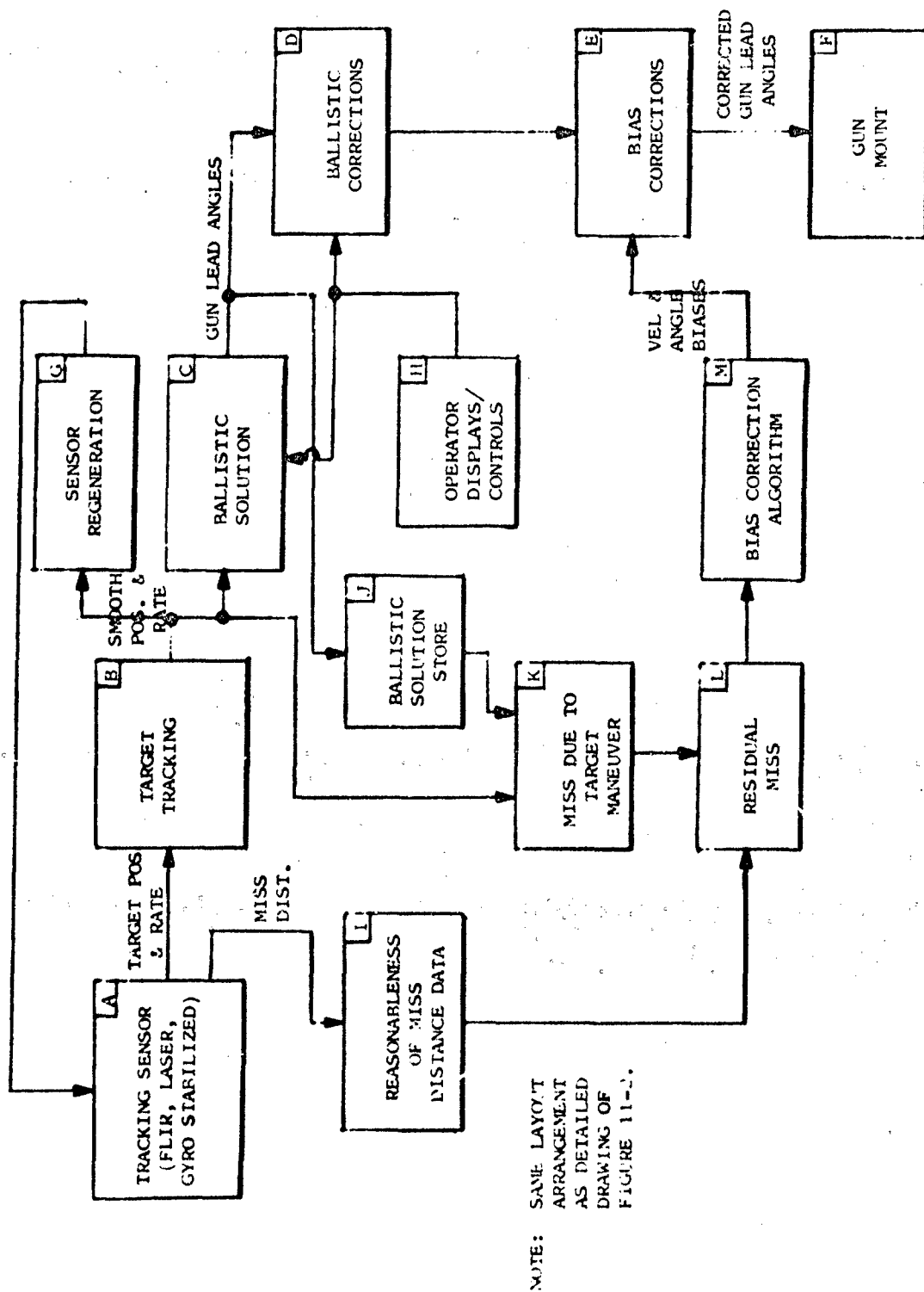
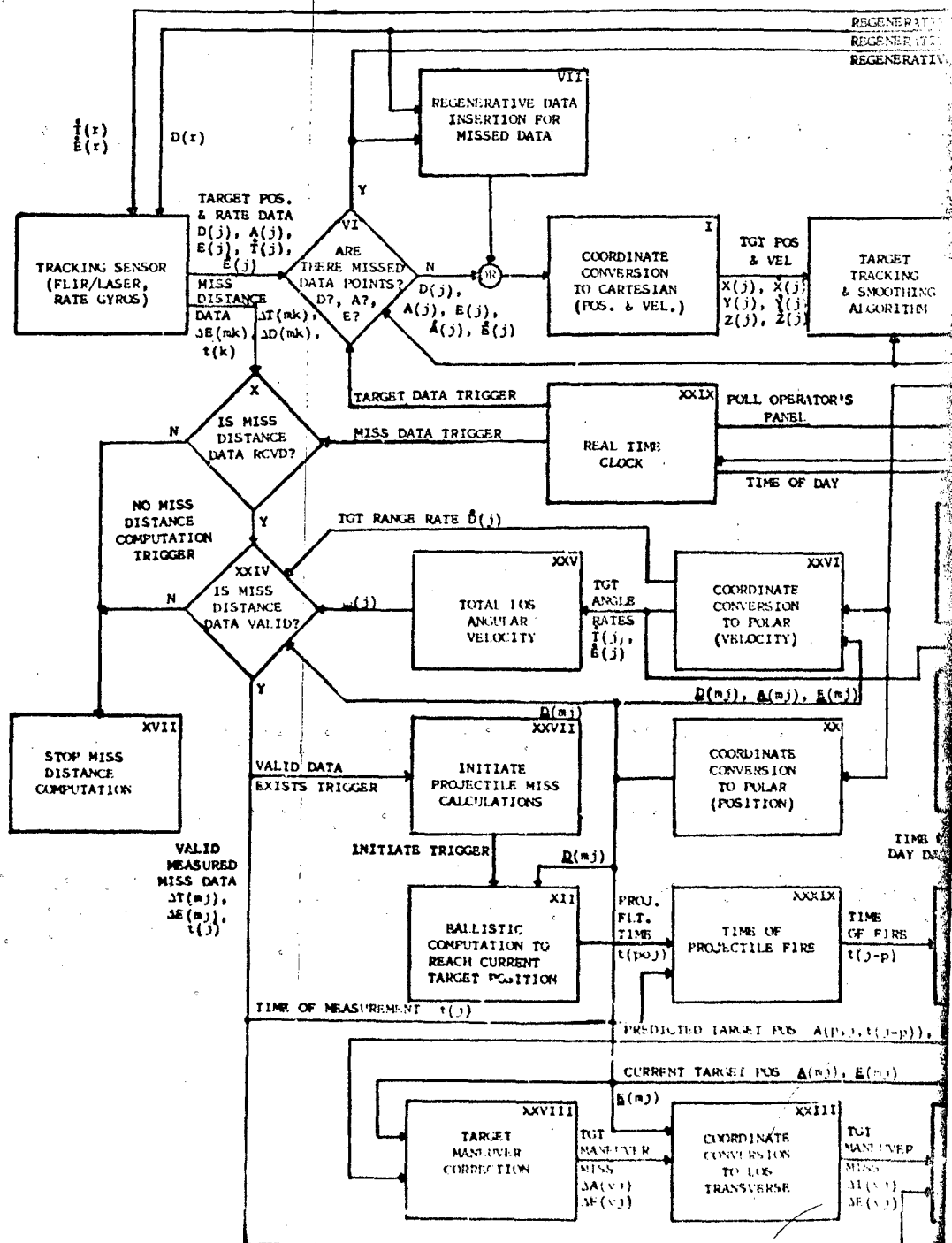


Figure 11-1. Simplified AFAADS Real Time Fire Control Block Diagram





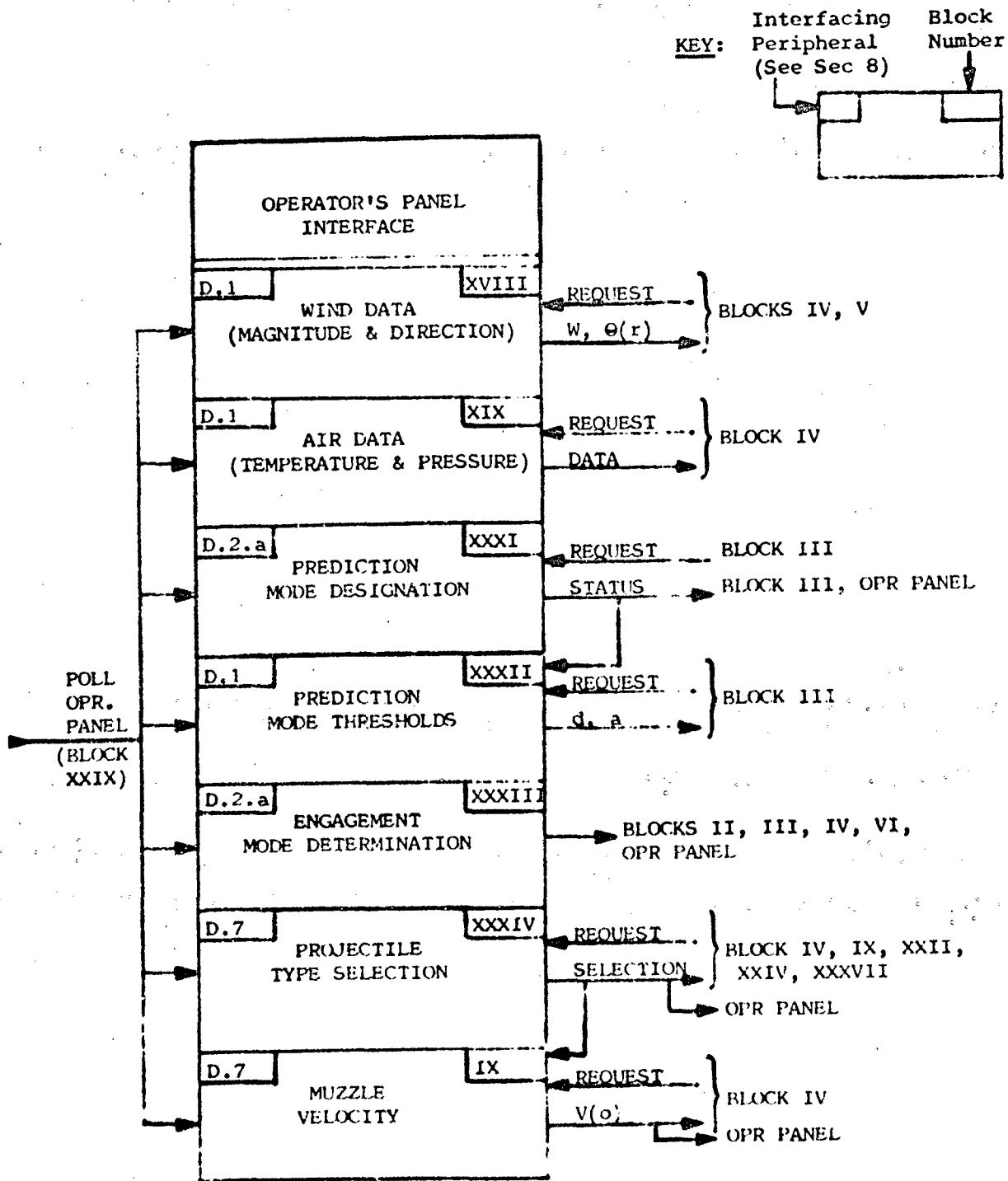


Figure 11-3. Operator's Panel Fire Control Interface Block Diagram

TABLE 11-1. Correlation Between Fire Control Block Diagrams

Figure 11-1 <u>Simplified Block Diagram</u>	Figure 11-2 <u>Detailed Block Diagram</u>
A Tracking Sensor	Tracking Sensor
B Target Tracking	I, II, VI, VII, XI
C Ballistic Solution	III, IV
D Ballistic Corrections	V, XXXV, XXXVI, XL, Vehicle Orientation Sensors
E Bias Corrections	XXII, XXXVII
F Gun Mount	Gun Mount
G Sensor Regeneration	VIII
H Operator Displays/Controls	IX, XVIII, XIX, XXXI, XXXII, XXXIII, XXXIV, Operator's Panel
I Reasonableness of Miss Distance Data	X, XVII, XXIV, XXV, XXVI
J Ballistic Solution Store	XIII, XXX
K Miss Due To Target Maneuver	XII, XIV, XV, XX, XXVII, XXVIII, XXXIX
L Residual Miss	XXIII, XXXVIII
M Bias Correction Algorithm	XVI, XXI
- Clock	XXIX

The algebraic symbols used for the input, algorithm, output, and the detailed block diagram (Figures 11-2 and 11-3) are identical for the same signal. Also identical titles are used throughout. The peripheral input/output devices use the same nomenclature introduced in Section 8.

Block I: Coordinate Conversion to Cartesian (Position and Velocity)

1. Function:
  - a. Find slant range rate.
  - b. Convert polar coordinates<sup>+</sup> and their rates to Cartesian coordinates and their rates<sup>+</sup>.
  - c. During the target acquisition phase, additional computations must be performed prior to the above two coordinate conversion functions. These additional computations deal with any missed range, azimuth, elevation, and range rate data. The target acquisition phase is the period following FLIR and laser detection when the tracking filter in the computer is receiving its first one second of data (eleven data points).
2. Frequency: Every 0.1 second
3. Inputs:
  - 3.1 Target Position Data: From Block VI<sup>\*</sup>

---

<sup>+</sup> All coordinates except as specifically noted are vehicle coordinates, not earth coordinates.

<sup>\*</sup> With some possibility of slant range (D) and/or elevation and azimuth (E or A) being missed, these values may have been computed -- not measured -- from previous scan. See Block VII.

$D(j)$  = Raw target range as measured by the laser at time  $j$ , positive out.

Magnitude: 100m to 10,000m

Accuracy: 0.25m

$A(j)$  = Raw target azimuth as measured by the FLIR at time  $j$ , positive clockwise from front of vehicle.

Magnitude: 0 deg to 360 deg.

Accuracy: 0.25 mil.

$E(j)$  = Raw target elevation data as measured by the FLIR at time  $j$ , positive up from plane of AFAADS vehicle.

Magnitude: -5 deg to + 85 deg.

Accuracy: 0.25 mil.

### 3.2 Target Angular Data: From Block VI.

$\dot{T}(j)$  = Transverse target rate as measured by transverse rate gyro at time  $j$ , positive clockwise.

Magnitude: -90 deg/sec to + 90 deg/sec.

Accuracy: 0.5 deg/sec.

$\dot{E}(j)$  = Elevation target rate as measured by elevation rate gyro at time  $j$ , positive up.

Magnitude: -25 deg/sec to + 25 deg/sec.

Accuracy: 0.2 deg/sec.

## 4. Constants:

a. During target acquisition phase, temporary storage is required as follows: Single value storage for the latest received raw data values of  $D(j)$ ,  $A(j)$ ,  $E(j)$ ,  $\dot{D}(j)$ ,  $\dot{T}(j)$ , and  $\dot{E}(j)$  and a time tag for each.

b. At all times, last target range  $D(j-1)$ .

## 5. Operations Performed:

### 5.1 Acquisition Phase Only

These operations are performed prior to the standard coordinate conversion operations of Item 5.2 below and are carried out only during the target acquisition phase. In addition, they differ (a) for the data received

immediately following initial target detection and (b) for the subsequently received data during the acquisition phase.

#### 5.1.1 At Initial Target Detection

When the target is first detected, the five quantities  $D(j)$ ,  $A(j)$ ,  $E(j)$ ,  $\dot{I}(j)$ , and  $\dot{E}(j)$  are all received. These are all stored and time tagged. Only the position coordinates  $D(j)$ ,  $A(j)$ , and  $E(j)$  are forwarded to the standard function of Item 5.2 below. (Thus, only the first position point (not velocity) in the tracking filter is filled.)

#### 5.1.2 On Subsequent Sensor Scans

The routine for subsequent scans during target acquisition is shown in Figure 11-4. The routine first determines if all five input quantities from the sensors are present. If they are, a second question is asked: is this set of data points the second set on a target? If so, a special routine is required to input first position rate data since range rate,  $\dot{D}(j)$ , is always a derived quantity. The second and subsequent sets of data are then processed in the same manner; namely, to update the storage of last recorded values and to pass this same data to the coordinate conversion function of Item 5.2 and hence to the tracking filter (Block II).

The remainder of the logic deals with the processing of missed data in range, azimuth or elevation. In each case, extrapolation from the previous value provides an initial input value.

#### 5.2 Both Acquisition and Tracking Phases

The following functions are performed during both the acquisition phase and the subsequent tracking phase.

Let  $j$  designate this scan, and  $j-1$  the immediately preceding scan.

Compute the following quantities:

$$\dot{D}(j) = [D(j) - D(j-1)] / \Delta t$$

where  $\Delta t = 0.1$  sec time between samples

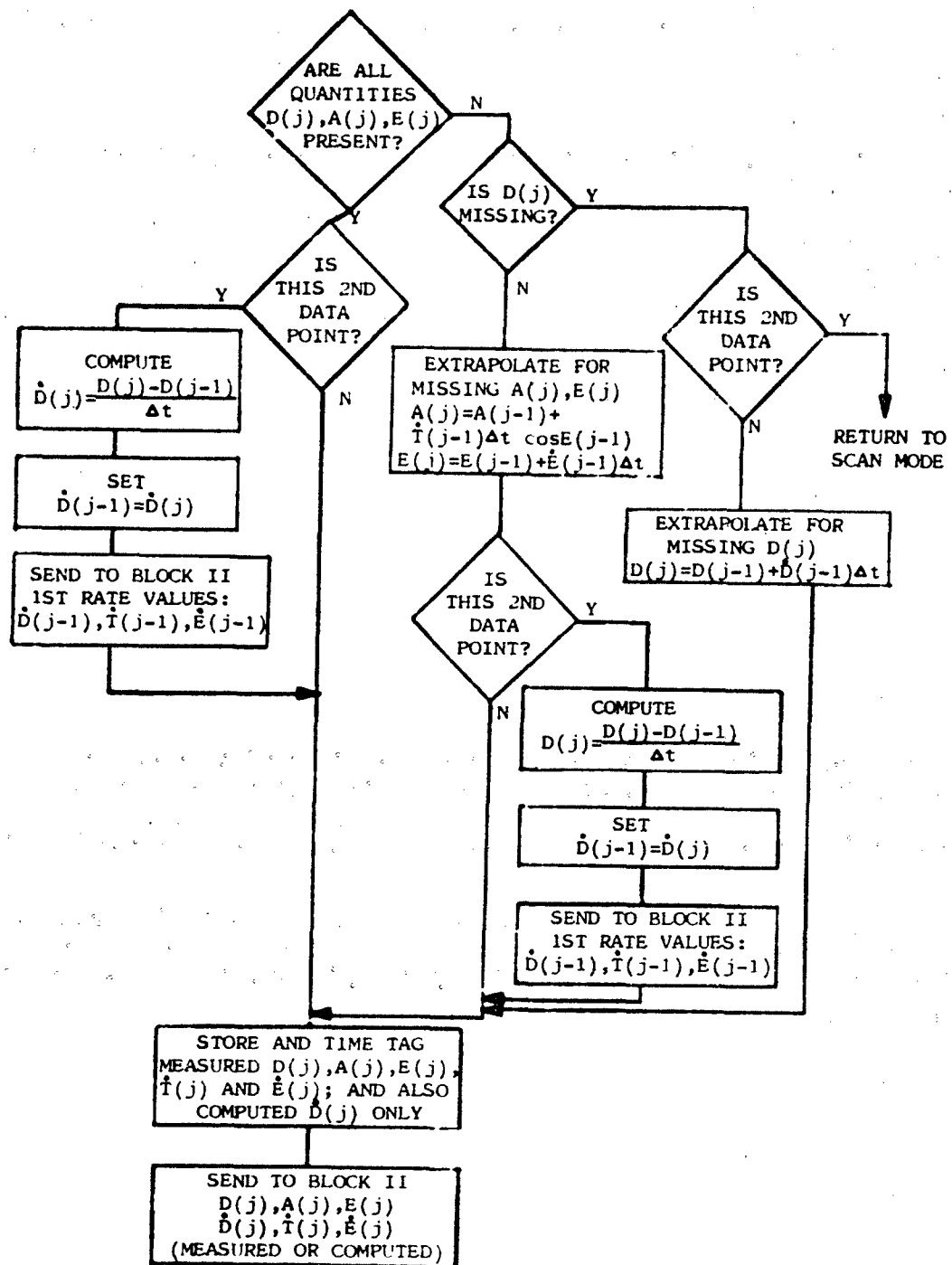


Figure 11-4. Portion of the Target Acquisition Routine

$$X(j) = D(j) \cos E(j) \sin A(j)$$

$$Y(j) = D(j) \cos E(j) \cos A(j)$$

$$Z(j) = D(j) \sin E(j)$$

$$\dot{X}(j) = [\dot{D}(j) \cos E(j) - \dot{E}(j) D(j) \sin E(j)] \sin A(j) + \dot{T}(j) D(j) \cos A(j)$$

$$\dot{Y}(j) = [\dot{D}(j) \cos E(j) - \dot{E}(j) D(j) \sin E(j)] \cos A(j) - \dot{T}(j) D(j) \cos A(j)$$

$$\dot{Z}(j) = \dot{D}(j) \sin E(j) + \dot{E}(j) D(j) \cos E(j)$$

## 6. Output:

### 6.1 Target Position:

To Block II, "Tracking and Smoothing Block," go present target position and rates in Cartesian coordinates. The values above have not been smoothed, nor have accelerations been computed.

$X(j)$  = Cross position of target at time  $t(j)$ , positive to right side  
Magnitude: -10,000m to +10,000m

Accuracy: \_\_\_\_\_ m

$Y(j)$  = Lengthwise position of target at time  $t(j)$ , positive forward  
Magnitude: -10,000m to +10,000m

Accuracy: \_\_\_\_\_ m

$Z(j)$  = Altitude of target above AFAADS vehicle at time  $t(j)$ , positive up (Negative value possible due to vehicle being parked on steep hillside)

Magnitude: -900m to +8,000 m

Accuracy: \_\_\_\_\_ m

### 6.2 Target Velocity:

To Block II

$\dot{X}(j)$  = Cross rate of target relative to AFAADS at time  $t(j)$ , right is positive

Magnitude: -400m/sec to +400m/sec

Accuracy: \_\_\_\_\_ m/sec

$\dot{Y}(j)$  - Lengthwise rate of target at time  $t(j)$ , positive forward.

Magnitude: -400m/sec to +400m/sec

Accuracy: \_\_\_\_\_ m/sec

$\dot{Z}(j)$  = Climb rate of target at time  $t(j)$ , positive up.

Magnitude: -400m/sec to +400m/sec

Accuracy: \_\_\_\_\_ m/sec

7. Comments:

It is hoped no accuracy will be lost in summing and multiplying values. Also, this block requires slightly more accurate cosine and sine routines than the angles (E and A) themselves, thereby insuring better accuracy.

Block II: Target Tracking and Smoothing Algorithm

1. Function: To develop the best estimate of the target's position, velocity, and acceleration from the sensor position and rate data. More specifically to smooth the raw-position and velocity data as given in Cartesian Coordinates into smoothed position, velocity, and acceleration data, still in Cartesian Coordinates. The tracking filter is of finite length and the output is the best estimate of the last measured target position and rate.

2. Frequency: Once every 0.1 second.

3. Inputs:

3.1 Sensor Data: From Block I.

$X(j)$  = Raw cross position of target relative to AFAADS vehicle at time  $t(j)$ ; right side is positive.

Magnitude: -10,000m to +10,000m.

Accuracy: \_\_\_\_\_ m.

$Y(j)$  = Raw lengthwise position of target relative to AFAADS vehicle at time  $t(j)$ ; positive forward.

Magnitude: -10,000m to +10,000m.

Accuracy: \_\_\_\_\_ m

$Z(j)$  = Raw altitude of target relative to plane of AFAADS vehicle at time  $t(j)$ ; positive up.

Magnitude: -900m to +8,000m.

Accuracy: \_\_\_\_\_ m.

- $\dot{X}(j)$  = Raw cross rate of target relative to AFAADS vehicle at time  $t(j)$ ; right side is positive.  
 Magnitude: -400m/sec to +400m/sec.  
 Accuracy: \_\_\_\_\_ m/sec.
- $\dot{Y}(j)$  = Raw lengthwise rate of target relative to AFAADS vehicle at time  $t(j)$ ; positive forward.  
 Magnitude: -400m/sec to +400m/sec.  
 Accuracy: \_\_\_\_\_ m/sec.
- $\dot{Z}(j)$  = Raw climb rate of target relative to AFAADS vehicle at time  $t(j)$ ; positive up.  
 Magnitude: -400m/sec to +400m/sec.  
 Accuracy: \_\_\_\_\_ m/sec.

- 3.2 Operating Mode Status: From Block XXXIII, status word indicating the operating mode to be used. Only the Semi-Automatic Fire Control and Test Modes will first clear and then activate the tracking filter
4. Constants: Algorithm smoothing constants or weighting constants for position, velocity, and acceleration smoothing. These are the specific constants for a 0.1 second data period and a 1.0 second filter; the values used in the AFAADS computer concepts analysis.

	<u>Data Interval</u>	<u>Weight</u>		
		<u>Position</u>	<u>Velocity</u>	<u>Acceleration</u>
	j	7/22	7/22	5/11
	j-1	6/22	6/22	4/11
	j-2	5/22	5/22	3/11
	j-3	4/22	4/22	2/11
	j-4	3/22	3/22	1/11
	j-5	2/22	2/22	0
	j-6	1/22	1/22	-1/11
	j-7	0	0	-2/11
	j-8	-1/22	-1/22	-3/11
	j-9	-2/22	-2/22	-4/11
	j-10	-3/22	-3/22	-5/11

## 5. Operations Performed

For a 0.1 second data rate and a 1.0 second filter the smoothing algorithms are:

### 5.1 Position Smoothing

Algorithm uses the position weighting constants of Item 4 above and the raw position data inputs from the last 11 input intervals. Output is the smoothed or best estimate of the target's position at the last measured time  $t(j)$ .

$$\underline{X}(j) = [7X(j)+6X(j-1)+5X(j-2)+4X(j-3)+3X(j-4)+2X(j-5) \\ +X(j-6)-X(j-8)-2X(j-9)-3X(j-10)]/22$$

$$\underline{Y}(j) = [7Y(j)+6Y(j-1)+5Y(j-2)+4Y(j-3)+3Y(j-4)+2Y(j-5) \\ +Y(j-6)-Y(j-8)-2Y(j-9)-3Y(j-10)]/22$$

$$\underline{Z}(j) = [7Z(j)+6Z(j-1)+5Z(j-2)+4Z(j-3)+3Z(j-4)+2Z(j-5) \\ +Z(j-6)-Z(j-8)-2Z(j-9)-3Z(j-10)]/22$$

### 5.2 Velocity Smoothing

Algorithm uses the velocity weighting constants of Item 4 and the raw velocity data inputs from the last 11 input intervals. Output is the smoothed or best estimate of the target's velocity at the last measured time  $t(j)$ .

$$\dot{\underline{X}}(j) = [7\dot{X}(j)+6\dot{X}(j-1)+5\dot{X}(j-2)+4\dot{X}(j-3)+3\dot{X}(j-4) \\ +2\dot{X}(j-5)+\dot{X}(j-6)-\dot{X}(j-8)-2\dot{X}(j-9)-3\dot{X}(j-10)]/22$$

$$\dot{\underline{Y}}(j) = [7\dot{Y}(j)+6\dot{Y}(j-1)+5\dot{Y}(j-2)+4\dot{Y}(j-3)+3\dot{Y}(j-4)+2\dot{Y}(j-5) \\ +\dot{Y}(j-6)-\dot{Y}(j-8)-2\dot{Y}(j-9)-3\dot{Y}(j-10)]/22$$

$$\dot{\underline{Z}}(j) = [7\dot{Z}(j)+6\dot{Z}(j-1)+5\dot{Z}(j-2)+4\dot{Z}(j-3)+3\dot{Z}(j-4) \\ +2\dot{Z}(j-5)+\dot{Z}(j-6)-\dot{Z}(j-8)-2\dot{Z}(j-9)-3\dot{Z}(j-10)]/22$$

### 5.3 Acceleration Smoothing

Algorithm uses the acceleration weighting constants of Item 4 and the raw velocity data inputs from the last 11 input intervals. Output is the smoothed or best estimate of the target's acceleration at the last measured time  $t(j)$ .

$$\ddot{X}(j) = [\dot{5X}(j) + 4\dot{X}(j-1) + 3\dot{X}(j-2) + 2\dot{X}(j-3) + \dot{X}(j-4) - \dot{X}(j-6) - 2\dot{X}(j-7) - 3\dot{X}(j-8) - 4\dot{X}(j-9) - 5\dot{X}(j-10)]/11$$

$$\ddot{Y}(j) = [\dot{5Y}(j) + 4\dot{Y}(j-1) + 3\dot{Y}(j-2) + 2\dot{Y}(j-3) + \dot{Y}(j-4) - \dot{Y}(j-6) - 2\dot{Y}(j-7) - 3\dot{Y}(j-8) - 4\dot{Y}(j-9) - 5\dot{Y}(j-10)]/11$$

$$\ddot{Z}(j) = [\dot{5Z}(j) + 4\dot{Z}(j-1) + 3\dot{Z}(j-2) + 2\dot{Z}(j-3) + \dot{Z}(j-4) - \dot{Z}(j-6) - 2\dot{Z}(j-7) - 3\dot{Z}(j-8) - 4\dot{Z}(j-9) - 5\dot{Z}(j-10)]/11$$

## 6. Outputs:

### 6.1 Target Position Data: To Blocks III, VIII, XI, XX, and XXVI.

$\underline{X}(j)$  = Smoothed cross position of the target relative to the AFAADS vehicle at the last measured time  $t(j)$ ; right side is positive.

Magnitude: -10,000m to +10,000m

Accuracy: 0.564 of raw input position data error.

$\underline{Y}(j)$  = Smoothed lengthwise position of the target relative to the AFAADS vehicle at the last measured time  $t(j)$ ; forward is positive.

Magnitude: -10,000m to +10,000m.

Accuracy: 0.564 of raw input position data error.

$\underline{Z}(j)$  = Smoothed altitude of target relative to AFAADS vehicle at the last measured time  $t(j)$ ; up is positive.

Magnitude: -900m to +8,000m.

Accuracy: 0.564 of raw input position data error.

### 6.2 Target Velocity Data: To Blocks III, VIII, XI, and XXVI.

$\dot{\underline{X}}(j)$  = Smoothed cross velocity of the target relative to the AFAADS vehicle at the last measured time  $t(j)$ ; right side is positive.

Magnitude: -400m/sec to +400m/sec.

Accuracy: 0.564 of raw input velocity data error.

$\dot{\underline{Y}}(j)$  = Smoothed lengthwise velocity of the target relative to the AFAADS vehicle at the last measured time  $t(j)$ ; forward is positive.

Magnitude: -400m/sec to +400m/sec.

Accuracy: 0.564 of raw input velocity data error.

$\dot{Z}(j)$  = Smoothed climb rate of target relative to AFAADS vehicle at the last measured time  $t(j)$ ; up is positive.

Magnitude: -400m/sec to +400m/sec.

Accuracy: 0.564 of raw input velocity data error.

6.3 Target Acceleration Data: To Blocks III (and XI as a growth feature)

$\ddot{X}(j)$  = Smoothed cross acceleration of the target relative to the AFAADS vehicle at the last measured time  $t(j)$ ; right side is positive.

Magnitude: -200m/sec<sup>2</sup> to +200m/sec<sup>2</sup>.

Accuracy: 0.955 of raw input velocity data error.

$\ddot{Y}(j)$  = Smoothed lengthwise acceleration of the target relative to the AFAADS vehicle at the last measured time  $t(j)$ ; forward is positive.

Magnitude: -200m/sec<sup>2</sup> to +200m/sec<sup>2</sup>

Accuracy: 0.955 of raw input velocity data error.

$\ddot{Z}(j)$  = Smoothed vertical acceleration of the target relative to the AFAADS vehicle at the last measured time  $t(j)$ ; up is positive.

Magnitude: -200m/sec<sup>2</sup> to +200m/sec<sup>2</sup>.

Accuracy: 0.955 of raw input velocity data error.

7. Comments: None.

8. References: Appendix B

Block III. Prediction Module and

Block IV. Ballistic Computation

Note: Although from a system's orientation these blocks should be discussed separately, the actual logic involved is not so easily separated.

1. Function: Basically, the two blocks involve:

- (a) Computing the predicted target position, and
- (b) Computing the projectile impact point such that the projectile will impact the target, i. e., be at the same range at the same time. In addition, the required algorithms also involve initialization, testing, and coordinate conversions.

Following each target position update (Block II), the following steps are carried out:

Block III

- (a) Find prediction mode desired, either based on automatic logic or in response to operator-ordered prediction (see Block XXXI).

Three types of prediction are instrumented. Possible combinations are:

- (1) Linear prediction, either automatic or operator-ordered.
- (2) Constant turn or acceleration, either automatic or operator-ordered, as a correction to linear prediction.
- (3) Dive or conservation of energy prediction, either automatic or operator-ordered, also as a correction to linear prediction.
- (4) Both (2) and (3). The logic controlling actions by the operator for the above choices are discussed in Section 7 and in Blocks XXXI and XXXII of this subsection.

In addition to the above, the prediction mode choices may grow to include:

- (5) Defense of a known point, operator-ordered only. Determination of the appropriate prediction mode results in the generation of a prediction word.

- (b) If this is a new target, use the prediction word to estimate the initial time of flight; or, if this is an old target, use the time of flight stored at the end of the previous processing; i. e., the processing following the last previous target position update. In either case, this time is called the predicted time of flight.
- (c) Use the predicted time of flight to compute the predicted target position in Cartesian coordinates using the appropriate prediction modes as determined above.
- (d) Convert to slant range.

#### Block IV

- (e) Using the ballistic equation, compute the ballistic time of flight using the slant range predicted in (d) above.
- (f) Compute the difference between the predicted time of flight (value used in Step (c)) and the ballistic time of flight (value determined in Step (e)).
- (g) If the difference is less than 0.005 second, go to (h) below. Otherwise, go to (c) above using the ballistic time of flight computed in Step (e) as the new predicted time of flight. Successive iterations of Steps (c), (d), (e), (f), and (g) will result in convergence to within 0.005 second.
- (h) Save the ballistic time computed in Step (e) as the final time of flight. This is also the time of flight used in Step (b), 0.1 second later when the next target position update is computed.
- (i) Compute the Cartesian coordinate position of the impact point based upon this time.
- (j) Convert the Cartesian coordinates to polar coordinates.

The predicted impact point and time-of-flight are the output results from these blocks.

- 2. Frequency: Every scan interval or 0.1 second.

### 3. Inputs

#### 3.1 Current Smoothed Target Position: From Block II

$\underline{X}(j)$  = Cross position of target relative to AFAADS vehicle, positive to right.

Magnitude: -10,000m to +10,000m

Accuracy: \_\_\_\_\_ m

$\underline{Y}(j)$  = Lengthwise position of target relative to AFAADS vehicle, positive forward

Magnitude: -10,000m to +10,000m

Accuracy: \_\_\_\_\_ m

$\underline{Z}(j)$  = Altitude of target above AFAADS vehicle, positive up.

Magnitude: -900m to +8,000m

Accuracy: \_\_\_\_\_ m.

#### 3.2 Current Smoothed Target Velocity: From Block II

$\dot{\underline{X}}(j)$  = Cross velocity of target relative to AFAADS, positive to right.

Magnitude: -400m/sec to +400m/sec

Accuracy: \_\_\_\_\_ m/sec

$\dot{\underline{Y}}(j)$  = Lengthwise acceleration of target relative to AFAADS, positive forward.

Magnitude: -400m/sec to +400m/sec

Accuracy: \_\_\_\_\_ m/sec

$\dot{\underline{Z}}(j)$  = Vertical velocity of target relative to AFAADS, positive up

Magnitude: -400m/sec to +400m/sec

Accuracy: \_\_\_\_\_ m/sec.

#### 3.3 Current Smoothed Target Acceleration: From Block II

$\ddot{\underline{X}}(j)$  = Cross acceleration of target relative to AFAADS, positive to right.

Magnitude: -200m/sec<sup>2</sup> to +200m/sec<sup>2</sup>

Accuracy: \_\_\_\_\_ m/sec<sup>2</sup>

$\ddot{\underline{Y}}(j)$  = Lengthwise acceleration of target relative to AFAADS, positive forward.

Magnitude: -200m/sec<sup>2</sup> to +200m/sec<sup>2</sup>

Accuracy: \_\_\_\_\_ m/sec<sup>2</sup>

$\ddot{Z}(j)$  = Vertical acceleration of target relative to AFAADS, positive up.  
Magnitude:  $-200\text{m/sec}^2$  to  $200\text{m/sec}^2$   
Accuracy: \_\_\_\_\_  $\text{m/sec}^2$

- 3.4 Muzzle Velocity: From Block IX during Initialization Mode, the Operator entered or preset value of muzzle velocity for use in correcting the coefficients in the ballistic equation.

$V(o)$  = Muzzle Velocity  
Preset value = (i) Full caliber:  $1200\text{m/sec}$   
(ii) Sub-caliber:  $1500\text{m/sec}$   
User Magnitude:  $930\text{m/sec}$  to  $1570\text{m/sec}$   
Accuracy:  $10\text{ m/sec}$ .

- 3.5 Wind: From Block XVIII during the Initialization Mode, Operator entered or preset value of wind for correcting the coefficients in the ballistic equation.

$W$  = Magnitude of wind  
Preset value:  $0\text{m/sec}$   
User magnitude:  $0\text{m/sec}$  to  $30\text{m/sec}$  (non-zero value to be accepted only if direction is specified)  
Accuracy:  $1.0\text{m/sec}$

$\Theta(r)$  = Relative direction of wind toward AFAADS, measured clockwise from front of vehicle.  
Preset value:  $0\text{ deg}$ .  
User magnitude:  $0\text{ deg}$ . to  $360\text{ deg}$ .  
Accuracy:  $5\text{ deg}$ .

- 3.6 Air Data: From Block XIX during the Initialization Mode, Operator entered or preset value of air temperature and pressure for correcting the coefficients in the ballistic equation.

$T$  = Air temperature  
Preset value:  $70\text{ deg. F}$   
User magnitude:  $-20\text{ deg.F}$  to  $+120\text{ deg.F}$   
Accuracy:  $1\text{ deg.F}$

P = Air pressure  
Preset Value: 1013 mb  
User magnitude: 500 mb to 1040 mb.  
Accuracy: 1 mb.

3.7 Prediction Mode Designation: From Block XXXI, status words indicating the particular prediction modes that are available for use.

Modes:

Linear  
Dive (energy conservation)  
Turn (or acceleration)

Designation:

Automatic  
Operator designated.

3.8 Prediction Mode Thresholds: From Block XXXII, prediction mode thresholds to be used in the prediction mode logic or when a particular prediction mode is to be used. Values specified as Operator entered or preset during the Initialization Mode.

d = Dive angle threshold  
Preset value: 10 deg. absolute  
User magnitude: 2 deg. to 15 deg. absolute  
Accuracy: 1 deg.

a/g = Turn (acceleration) threshold  
Preset value: 0.3g absolute  
User magnitude: 0.2g to 1.0g absolute.

3.9 Operating Mode Status: From Block XXXIII, operating mode status word indicating which operating mode AFAADS is in; Semi-automatic Fire Control, Manual, Test, Standby, or Initialization.

3.10 Projectile Type: From Block XXXIV, projectile type status word indicating the selected projectile type (one of five).

4. Constants:

4.1 Initial Time-of-Flight Table: Tables of flight time to be used in the initialization of the algorithm. Actually two tables are required, one for standard caliber rounds and one for sub-caliber rounds. Each table will have five values, for a total of 10 entries.

4.2 Final Time of Flight: Store of the final time-of-flight obtained from a given iterative solution (Step e. in Item 1. above); namely,  $t(pj')$ .

4.3 Allowable Time-of-Flight Error: Constant for the maximum allowable time-of-flight error between the target and projectile.

Magnitude: 0.005 sec.

4.4 Coefficients for Ballistic Computation: The ballistics tables for the gun have been approximated by a fifth order polynomial of the form:

$$t(p) = a D(pj) + b D^2(pj) + c D^3(pj) + d D^4(pj) + e D^5(pj)$$

where  $D(pj)$  is the predicted range to the target at impact. The five coefficients  $a$ ,  $b$ ,  $c$ ,  $d$ , and  $e$  are a function of air temperature, air pressure, muzzle velocity, wind, and projectile type.

The values of these coefficients are computed during the Initialization Mode based upon Operator inputs. (Blocks XIX, IX, XVIII, and XXXIV.) They are then stored for use during ballistic computations in both the Semi-automatic Fire Control Mode and the Test Mode (Dynamic System Test). These may have to be stored as double precision numbers or as single precision numbers with scale factors in a second word.

4.4.1 Initialization Mode Corrections: The following outlines the muzzle velocity, air temperature, and air pressure corrections that are applied to the five ballistic coefficients during the Initialization Mode computations.

- (a) Correct Muzzle Velocity. From  $V(o)$ , the muzzle velocity (input Item 3.4), correct for air temperature,  $T$ , (input Item 3.6)  $\Delta V(o) = K_1 T + K_2 T^2$   
where  $K_1$  and  $K_2$  are two different constants for each of five projectile types (10 constants total)  
Then the corrected muzzle velocity is  $V(c) = V(o) + \Delta V(o)$   
This value is both used here and outputted to Block V.

- (b) Correct for Air Pressure: Correct last four coefficients, b, c, d, and e, for non-standard air pressure where  
 P is air pressure (input Item 3.6)  
 $K_3$  is a constant, different for each projectile type.

Correction is

$$\Delta b = K_3 p$$

$$\Delta c = K_3 p$$

$$\Delta d = K_3 p$$

$$\Delta e = K_3 p$$

- (c) Ballistic Coefficient Corrections: For each projectile type, let

$b(o), c(o), d(o), e(o)$  = Ballistic coefficients for  
 standard temperature, pressure,  
 and temperature.

$b(l), c(l), d(l), e(l)$  = Correction constants for non-  
 standard muzzle velocity.

Then the corrected ballistic coefficients are:

$$b = b(o) + \Delta b + b(l) V(c)$$

$$c = c(o) + \Delta c + c(l) V(c)$$

$$d = d(o) + \Delta d + d(l) V(c)$$

$$e = e(o) + \Delta e + e(l) V(c)$$

- (d) First Ballistic Coefficient: The first ballistic coefficient a is

$$a = 1/V(c)$$

- 4.4.2 Sample Ballistic Coefficient Values: An estimate of the magnitude of the five ballistic equation coefficients a, b, c, d, and e can be obtained from a fourth order polynomial approximation to the 35 mm Oerlikon ballistic tables for a 0 degree elevation target. The resultant equation is:

$$t(p) = 8.5106 \times 10^{-4} D(pj) + 1.2146 \times 10^{-7} D^2(pj) \\ - 9.13 \times 10^{-12} D^3(pj) + 4.69 \times 10^{-15} D^4(pj)$$

where:  $t(pj)$  = Time of flight in seconds

$D(pj)$  = Predicted target range in meters.

This equation has a maximum error of about 0.002 second out to 4,100m range, increasing to 0.01 second at 4,500m range. The 4,500m range value is taken to be point at which projectile velocity drops to near sonic velocity. To match the original tables to 0.005 second accuracy would require very slight changes in the coefficients given here and would, undoubtedly, require the addition of one more term to the polynomial, i. e. the 5th term.

Although not stated directly in the table, the following factors are assumed:

- 1) Wind velocity was zero
- 2) Projectile was standard armour-piercing high explosive round.
- 3) Muzzle velocity was approximately 1200m/sec.
- 4) Gun barrel was relatively new.
- 5) Powder temperature was at the standard temperature of 15 deg. C. (59 deg. F)
- 6) Air pressure was at standard sea level value 1013mb.

The Initialization Mode processing of the ballistic solution coefficients considers variations in all of the above parameters.

5. Operations Performed: The logic during the Semi-Automatic Fire Control or Test Mode is first presented, followed by a detailed listing of the specific equations used.

5.1 Description of the Logic: The logic flow for the trajectory prediction and ballistic solution processing is given in Figure 11-5. It is keyed to the following description by the letter in the upper right corner of each logic flow block.

- (a) Upon receipt of the updated target position, velocity, and acceleration from Block II and upon recognition of being in either the Semi-Automatic Fire Control Mode or Test Mode (System Test), the prediction mode to be used is determined.

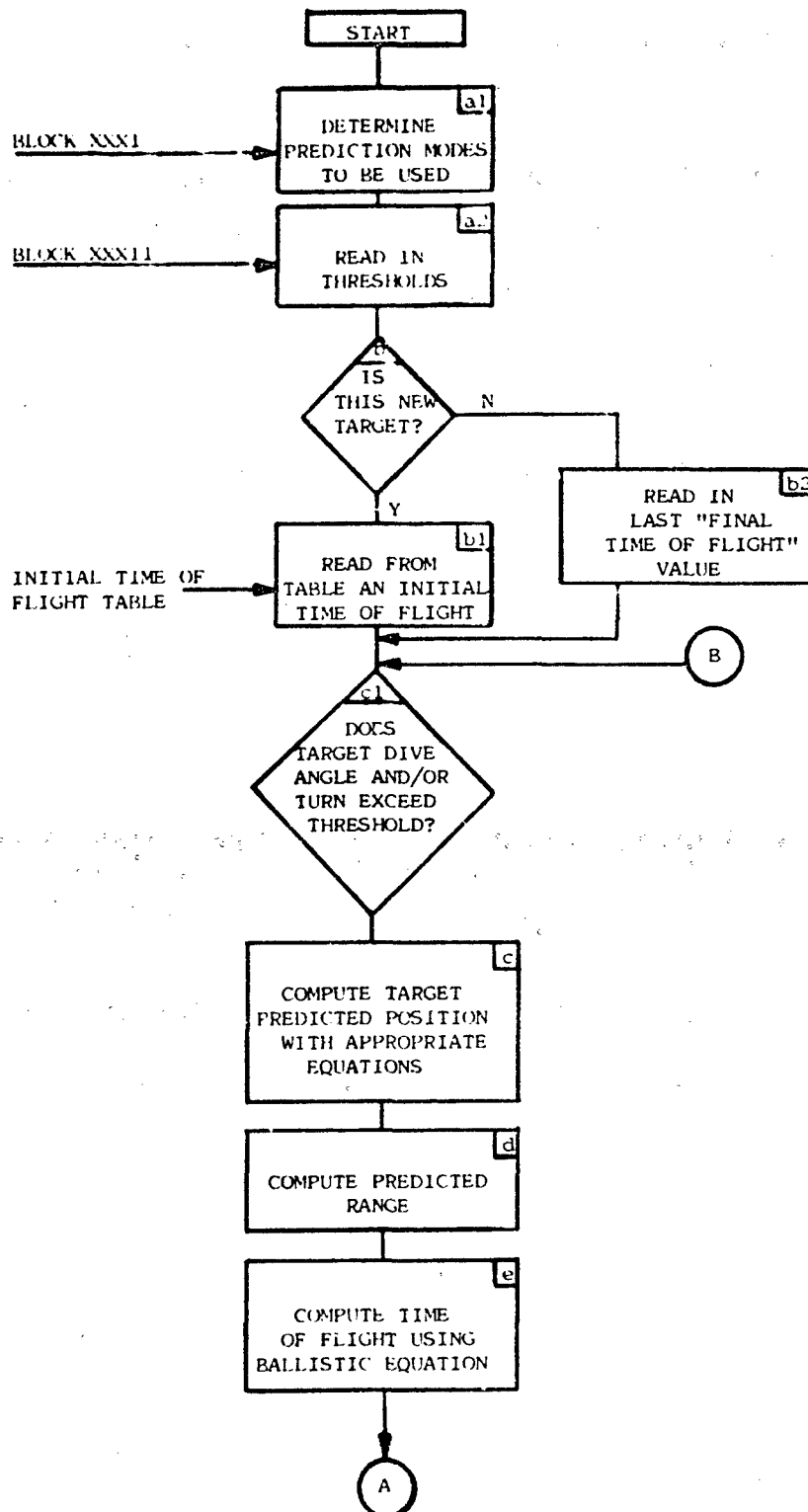


Figure 11-5. Target Prediction and Ballistic Solution Logic (Sheet 1 of 2)

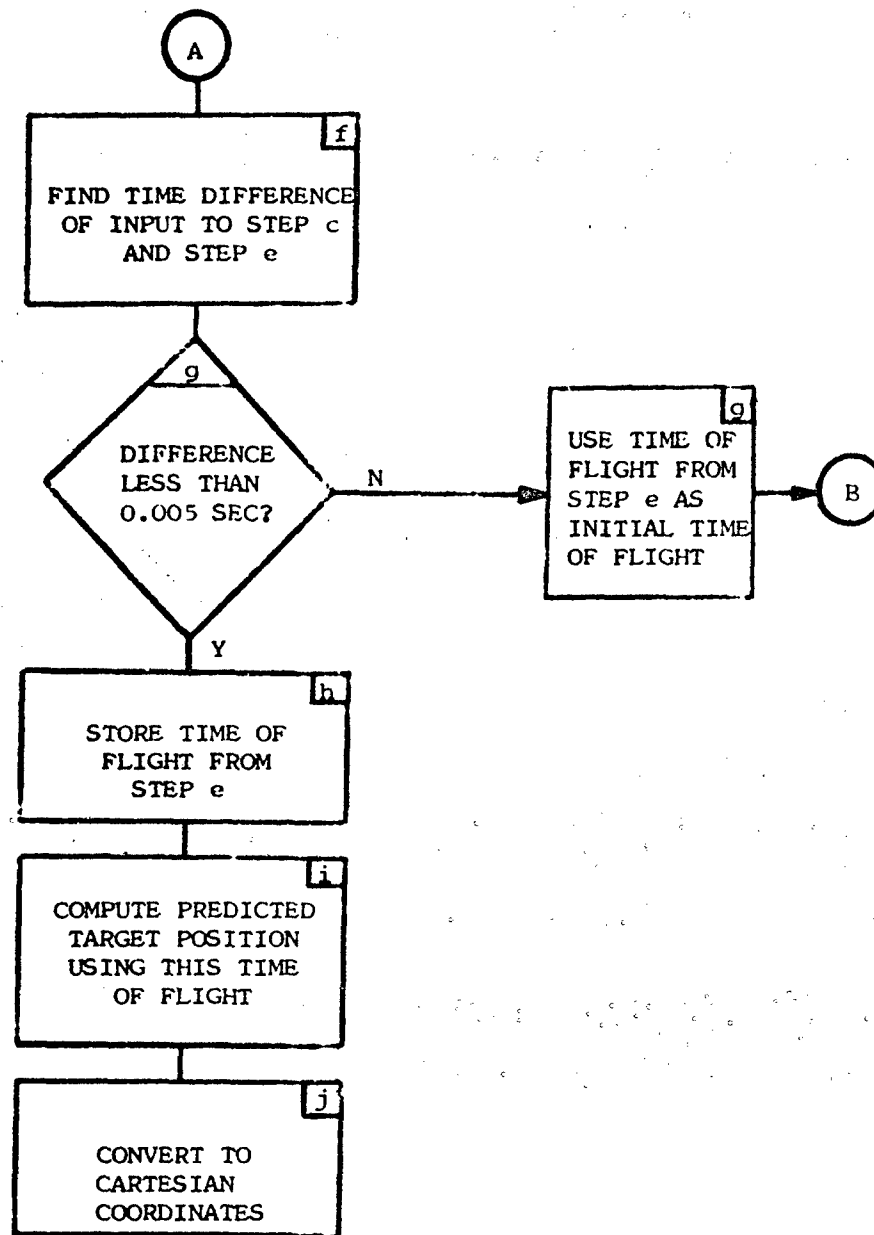


Figure 11-5. Target Prediction and Ballistic Solution Logic (Sheet 2 of 2)

- 1) Interrogate Block XXXI (Item 3.7 above) for the prediction modes to be used. This determines the applicable prediction equations.
  - 2) Based upon the prediction modes to be used, interrogate Block XXXII (Item 3.8 above) for the applicable thresholds.
- (b) Find predicted time of flight,  $t(pj')$
- 1) If this is the first time calculation on this target, use the appropriate value from table of Item 4.1 above.
  - 2) If previous calculations have been made on this target, use the last final time-of-flight value (Item 4.2 above).
- (c) Use the appropriate prediction equations from Step (a) to obtain the predicted position of the aircraft in Cartesian coordinates.
- 1) Determine if the target's climb (dive) angle and/or turning rate exceeds either or both the dive threshold and/or the turn threshold as entered in Step (a) 2) above.
  - 2) Based upon the threshold tests and the applicable prediction modes, determine the exact prediction equations to be used.
  - 3) Compute the predicted target position in Cartesian coordinates using these equations.
- (d) Find the predicted range from the predicted position in Cartesian coordinates.
- (e) Compute the ballistic time-of-flight using the fifth order polynomial in range.
- (f) Find the difference in time-of-flight between that used in Step (c) and that calculated in Step (e).
- (g) If the absolute value of the difference is less than 0.005 second (constant from Item 4.3 above), go to Step (h); otherwise go to Step (c) with a change in the predicted time-of-flight to the ballistic value determined in Step (e).
- (h) Save this ballistic time as the final time of flight by storing it in place of previous value. (See Item 4.2 above).
- (i) Compute final predicted Cartesian coordinate position using the final time of flight and the appropriate prediction equations.
- (j) Convert Cartesian coordinates to polar coordinates to end the processing.

5.2 Equations: Only certain of the steps in the logic flow of Item 5.1 above involve algebraic equations. These steps are described in detail below. The logic steps are self-evident and already completely described in Item 5.1 above.

Step c.1 Dive Angle Threshold

- (i) Compute target dive (climb) angle, using Item 3.2 inputs:

$$d(tj) = \tan^{-1} \left\{ \dot{Z}(j) / [\dot{X}(j)^2 + \dot{Y}(j)^2]^{1/2} \right\}$$

- (ii) Logical decision following computation: Is the absolute value of  $d(tj)$  greater than the dive angle threshold ( $d$ ) (Item 3.8)? If yes, then the dive or constant energy prediction term is to be used in the prediction mode equation. If no, this term is not to be used.

Step c.2 Constant Turn (Acceleration) Threshold

- (i) Compute target's radial acceleration in a horizontal turn, using Item 3.1 and 3.2 inputs:

$$a(tj) = [\dot{X}(j)^2 + \dot{Y}(j)^2] / [X(j)^2 + Y(j)^2]^{1/2}$$

- (ii) Logic decision following computation. Is the absolute value of  $a(tj)$  greater than the turn threshold ( $a$ ) (Item 3.8)? If yes, then the constant turn term is to be used in the prediction mode equation. If no, this term is not to be used.

Step c.3 Target Prediction

The predicted target position is the sum of the present position, linear prediction, dive prediction, and constant turn prediction. Note that constant turn prediction is only done in the horizontal coordinates X and Y, not in the vertical. The prediction equations are:

$$X(pj') = X(j) + \dot{X}(j) [t(pj') + T(s)/2]$$

$$-(g/2) [\dot{X}(j) \dot{Z}(j) / V^2(j)] [t^2(pj') + t(pj') T(s) + T^2(s)] \\ + (1/2) \dot{Y}(j) W(a_j) [t^2(pj') + t(pj') T(s) + T^2(s)]$$

$$Y(pj') = Y(j) + \dot{Y}(j) [t(pj') + T(s)/2]$$

$$-(g/2) [\dot{Y}(j) \dot{Z}(j) / V^2(j)] [t^2(pj') + t(pj') T(s) + T^2(s)] \\ + (1/2) \dot{X}(j) W(a_j) [t^2(pj') + t(pj') T(s) + T^2(s)]$$

$$Z(pj') = Z(j) + \dot{Z}(j) [t(pj') + T(s)/2]$$

$$-(g/2) [\dot{Z}^2(j) / V^2(j)] [t^2(pj') + t(pj') T(s) + T^2(s)]$$

where:  $t(pj')$  = Predicted time of flight

$T(s)$  = Sampling interval or 0.1 second

$g$  = Acceleration due to gravity

$v^2(j) = [\dot{x}^2(j) + \dot{y}^2(j)]^{1/2} / [x^2(j) + y^2(j)]^{1/2}$  = Angular velocity of aircraft in horizontal plane.

#### Step d: Predicted Target Range

The predicted range to the target is computed from the predicted target position developed in Step c.3.

$$D(pj') = [x^2(pj') + y^2(pj') + z^2(pj')]^{1/2}$$

#### Step e: Ballistic Time of Flight

Using the fifth order polynomial ballistic equation computed during the Initialization Mode (see Item 4.4), the ballistic time of flight

$$t(pj) = a D(pj') + b D^2(pj') + c D^3(pj') + d D^4(pj') + e D^5(pj')$$

where:  $D(pj')$  = Predicted target range from Step d

$a, b, c, d, e$  = Coefficients computed in Item 4.4.

#### Step f: Error in Predicted Time of Flight

Compute the difference between the predicted time of flight  $t(pj')$  (Step c) and the ballistic time of flight  $t(pj)$  (Step e).

$$E(j) = t(pj') - t(pj).$$

#### Step g. Convergence Threshold

Is the absolute value of  $E(j)$  less than the allowable time of flight error of 0.005 second (Item 4.3)? If no, the iteration of Steps c, d, e, f, and g must be repeated with the ballistic time of flight  $t(pj)$  designated as the new predicted time of flight  $t(pj')$ . If yes, proceed to Step h.

#### Step i. Final Predicted Target Position

Compute the final predicted target position using the same equations as in Step c.3 except the time is  $t(pj)$  instead of  $t(pj')$ . The resultant coordinates are  $X(pj)$ ,  $Y(pj)$ , and  $Z(pj)$ .

Step j. Coordinate Conversion to Polar Coordinates

The final predicted target position in Cartesian coordinates of Step i. are converted to polar coordinates by:

$$D(pj) = [X^2(pj) + Y^2(pj) + Z^2(pj)]^{1/2}$$

$$E(pj) = \sin^{-1} [Z(pj) / D(pj)]$$

$$A(pj) = \begin{cases} \sin^{-1} [X(pj) / D(pj) \cos E(pj)] \\ \text{or} \\ \cos^{-1} [Y(pj) / D(pj) \cos E(pj)] \end{cases}$$

6. Outputs:

6.1 Predicted Target Position (Cartesian Coordinates): To Block XL

$X(pj)$  = Predicted cross position of target at time of projectile impact, positive right.

Magnitude: -10,000m to 10,000m

Accuracy: \_\_\_\_\_ m (Note Oerlikon firing table accuracy is 5m)

$Y(pj)$  = Predicted lengthwise position of target, positive forward.

Magnitude: -10,000m to +10,000m

Accuracy: \_\_\_\_\_ m (Note Oerlikon firing table accuracy is 5m)

$Z(pj)$  = Predicted vertical position of target, positive up.

Magnitude: -900m to 8,000m

Accuracy: \_\_\_\_\_ m (Note Oerlikon firing table accuracy varies from 0.3m to 4.5m).

6.2 Predicted Target Range: To Block XL

$D(pj)$  = Predicted range to target at time of projectile impact.

Magnitude: 100m to 10,000m

Accuracy: \_\_\_\_\_ m

6.3 Predicted Target Angles: To Block XXX

A(pj) = Predicted target azimuth at time of projectile impact  
measured clockwise

Magnitude: 0 deg. to 360 deg.

Accuracy: \_\_\_\_\_ mils (Note: Oerlikon firing table accuracy  
is 1.5 mils)

E(pj) = Predicted target elevation, positive up.

Magnitude: -5 deg to +85 deg.

Accuracy: \_\_\_\_\_ mils (Note: Oerlikon firing table accuracy  
is 1.5 mils)

6.4 Predicted Time of Flight: To Blocks IV and XXX

t(pj) = Projectile time of flight

Magnitude: 0 sec to 10 sec.

Accuracy: 0.005 sec.

6.5 Corrected Muzzle Velocity: To Block V

V(c) = Corrected muzzle velocity (computed during Initialization Mode-  
see Item 4.4.1)

Magnitude: 930m/sec to 1570m/sec

Accuracy: 10m/sec

7. Comments: Accuracy of firing tables is three decimal digits. The accuracy of computations following this block are thus somewhat worse. With better ballistic tables, lesser errors will result.

8. References:
- a) Volume I
  - b) Reference 1, Section 4
  - c) Reference 2

Block V: Ballistic Corrections

1. Function: To compute the ballistic lead angle corrections for:

- a) gravity drop
- b) cross wind
- c) down wind

2. Frequency: Whenever gun orders are generated, i. e., every 0.1 second.

### 3. Inputs:

#### 3.1 Predicted Target Position in Polar Earth Coordinates: From Block XL

A(oj) = Predicted target azimuth relative to surface of the earth at the time of projectile impact; measure clockwise from front of AFAADS.

Magnitude: 0 deg to 360 deg.

Accuracy: \_\_\_\_\_ mils

E(oj) = Predicted target elevation relative to the surface of the earth, positive up.

Magnitude: 0 deg to 85 deg.

Accuracy: \_\_\_\_\_ mils

#### 3.2 Predicted Time of Flight: From Block IV

t(pj) = Predicted time-of-flight of the projectile

Magnitude: 0 to 10 sec.

Accuracy: 0.005 sec.

#### 3.3 Muzzle Velocity: From Block IV

V(c) = Corrected muzzle velocity

Magnitude: 930m/sec to 1570m/sec

Accuracy: 10m/sec.

#### 3.4 Wind: From Block XVIII

W = Magnitude of entered wind velocity

Magnitude: 0m/sec to 30m/sec (66 mph)

Accuracy: 1.0m/sec.

$\theta(r)$  = Relative direction of the wind toward AFAADS, measured clockwise from front of AFAADS vehicle.

Magnitude: 0 deg to 360 deg.

Accuracy: 5 deg.

### 4. Constants:

#### 4.1 Gravity Drop Correction:

g = Acceleration due to gravity

Magnitude: 9.8061m/sec

Accuracy: As required for negligible effect on computations.

$c_1$  = Constant, derived from Appendix F, Figure F-6  
Magnitude: 0.0465/sec.

4.2 Wind Corrections: Derived from Appendix G, Figure G-4

$k_1$  = 0.1153 milliradians/meter

$k_2$  = 0.0293/sec.

5. Operations Performed:

5.1 Gravity Drop Correction: (See Appendix F) Superelevation angle, positive up.

$$\phi(sj) = [g/2V(c)] [t(pj) + c_1 t(pj)^2] \cos E(oj)$$

5.2 Wind Correction:

5.2.1 Cross Wind: (See Appendix G)

Cross wind, positive when coming from left of target line of sight.

$$W(cj) = W \sin [A(oj) - \theta(r)]$$

Projectile drift, positive angle measured clockwise

$$L(j) = k_1 W(cj) t(pj) \sec E(oj) [1 - k_2 t(pj) \sin E(oj)]$$

5.2.2 Down Wind: (See Appendix C)

Down wind, positive away from AFAADS vehicle

$$W(dj) = -W \cos [A(oj) - \theta(r)]$$

Superelevation angle, positive up

$$\phi(wj) = -1.5 k_1 W(dj) t(pj) \tan E(oj) [1 - k_2 t(pj) \sin E(oj)]$$

5.3 Corrected Gun Angles:

$$A(goj) = A(oj) + L(j)$$

$$E(goj) = E(oj) + \phi(sj) + \phi(wj)$$

6. Outputs

6.1 Corrected Gun Angles in Earth Coordinates: To Block XXXV

$A(goj)$  = Ballistically corrected gun azimuth in earth coordinates, measured clockwise from front of vehicle.

Magnitude: 0 deg to 360 deg.

Accuracy: \_\_\_\_\_ mils.

E(goj) = Ballistically corrected gun elevation in earth coordinates,  
positive up.

Magnitude: 0 deg to 85 deg.

Accuracy: \_\_\_\_\_ mils.

7. Comments: None

8. References:

a) Appendix C

b) Appendix F

c) Appendix G

d) Reference 1, Volume I, paragraph 5.5.4 (p.5-53)

#### Block VI: Are There Missed Data Points?

1. Function:

- a) To determine if the sensors are providing range, azimuth, and elevation target data every look, and
- b) If not, to generate the regenerative triggers required so as to use extrapolated old target position data.

2. Frequency: Every 0.1 second

3. Inputs:

3.1 Target Data Trigger: From Block XXIX, a trigger is received every 0.1 second to read the range, azimuth, elevation, azimuth rate, and elevation rate sensor buffers.

3.2 Target Range Data: From the laser input buffer, peripheral device D3.a.2.

D(j) = Raw target range as measured by the laser at time j.

Magnitude: 100m to 10,000m

Accuracy: 0.25m

3.3 Target Angle Data: From the FLIR input buffer, peripheral device D3.a.1.

A(j) = Raw target azimuth as measured by the FLIR at time j, positive clockwise from front of AFAADS.

Magnitude: 0 deg to 360 deg.

Accuracy: 0.25 mil.

$E(j)$  = Raw target elevation as measured by the FLIR at time  $j$ , positive up.

Magnitude: -5 deg to +85 deg.

Accuracy: 0.25 mil.

- 3.4 Target Angular Rates: From the rate gyro input buffer, peripheral device 4.

$\dot{T}(j)$  = Raw slew rate of the transverse rate gyro, which for the tight servo system on AFAADS is a good measure of the target's transverse rate. Positive clockwise.

Magnitude: -90 deg/sec to +90 deg/sec.

Accuracy: 0.6% or 0.5 deg/sec.

$\dot{E}(j)$  = Raw slew rate of the elevation rate gyro, a good measure of the target's elevation rate, positive up.

Magnitude: -25 deg/sec to +25 deg/sec.

Accuracy: 0.6% or 0.2 deg/sec.

- 3.5 Operating Mode: From Block XXXIII a signal indicating which operating mode the AFAADS is now in (Semi-Automatic Fire Control, Manual, Standby, etc.)

4. Constants: Temporary storage in terms of clocks, counters, and flags are required for the following functions:

- a) During target acquisition to determine if sufficient target returns have been received over a one-second period to initiate automatic fire control computations.
- b) During target tracking to detect loss of tracking over a two-second period.

Specific counters or clocks are:

- 4.1 Acquisition Phase:

- a) Laser Counter or Flags: Record of number of range laser returns received during last 1.0 second (11 data points including current data point).
- b) FLIR Counter or Flags: Record of number of FLIR returns received during last 1.0 second (11 data points including current data point). Assumes either both azimuth and elevation data present or neither are available.

#### 4.2 Tracking Phase:

- a) Laser Clock or Counter: Time elapsed since last received laser range return. Maximum capacity of 2.0 seconds or 21 data returns.
- b) FLIR Clock or Counter: Time elapsed since last received FLIR returns. Maximum capacity of 2.0 seconds or 21 data returns.

5. Operations Performed: The specific operations performed by this block depend upon whether the system is in the Acquisition Phase or the Tracking Phase of the Semi-Automatic Fire Control Mode. Test Mode operations require additional or modified logic (TBD).

5.1 Acquisition Phase: Different logic is used during the Acquisition Phase for:

- a) Determining that a valid target is present, and
- b) Filling the tracking filter after the presence of a valid track has been established.

5.1.1 Initial Detection Routine: Used by the computer to determine whether a valid target return is being detected. This phase is entered following pushing SCAN by the operator.

The logic flow shown on Figure 11-6 is based upon the premise that both laser range data and FLIR azimuth and elevation data must be simultaneously present on the input buffers to establish a valid target. If data is present from only one sensor, target initialization will not start. It is assumed that the rate gyros will always provide good transverse and elevation rate data.

5.1.2 Initial Data Routine: This routine, shown on Figure 11-7, is used initially to fill the 11-point tracking filters of Block II. This is done on the criteria that there must be 10 good range values and 10 good angle values during the 1.0 second following initial detection before tracking can commence. If less than 10 good data points are received, the computer returns to the Initial Detection Routine of Item 5.1.1 above. After 10 or 11 good data points in the last 1.0 second are received, the computer changes to the Tracking Mode of Item 5.2 below.

5.2 Tracking Routine: This routine, shown in Figure 11-8, is used once target tracking has been established. It includes logic to determine if:

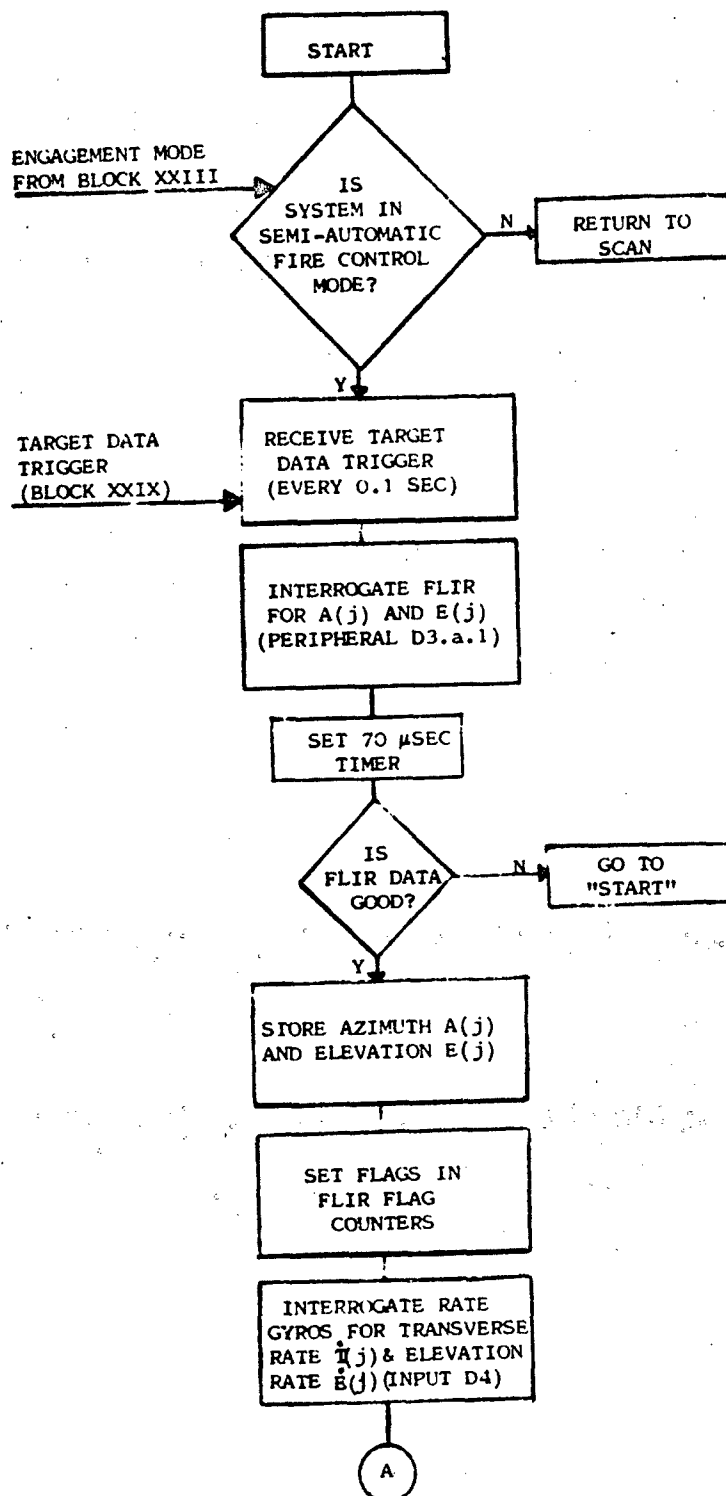


Figure 11-6. Initial Detection Routine  
(Sheet 1 of 2)

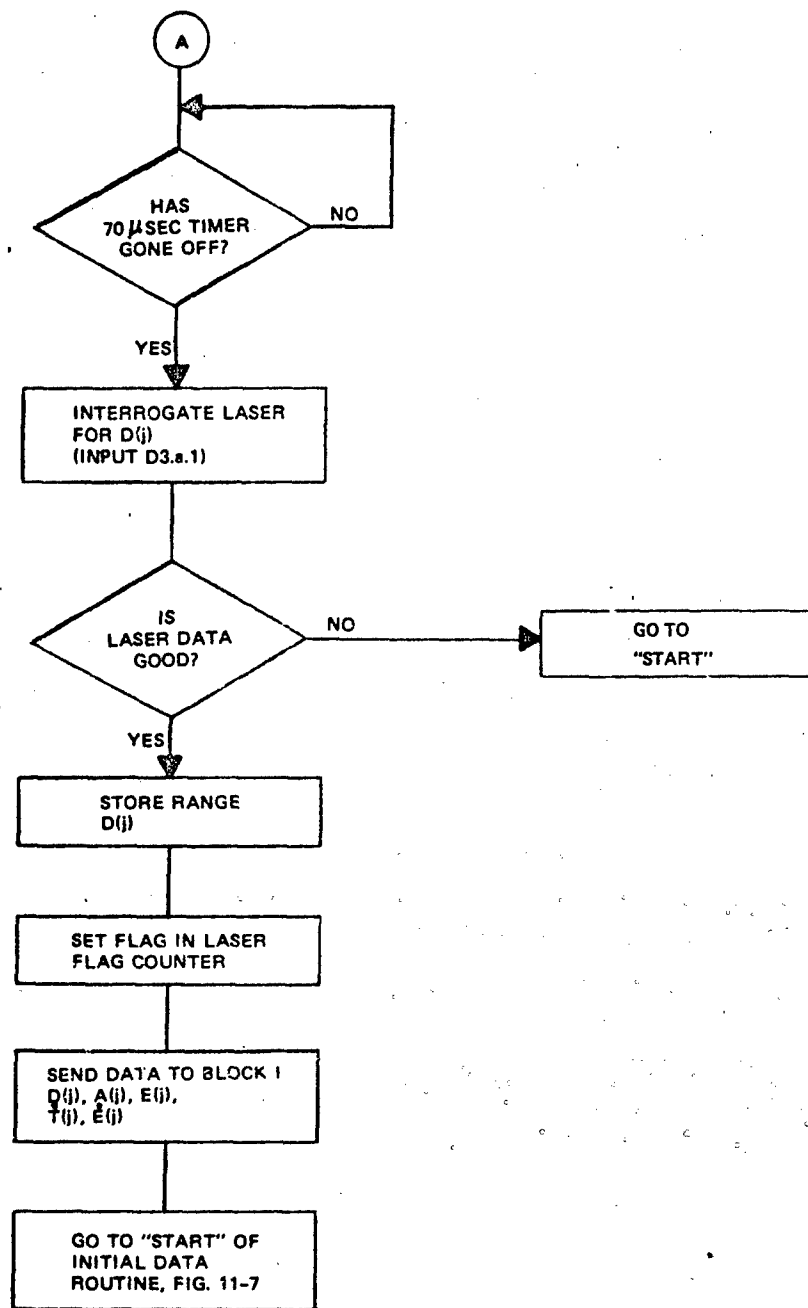


Figure 11-6. Initial Detection Routine (Sheet 2 of 2).

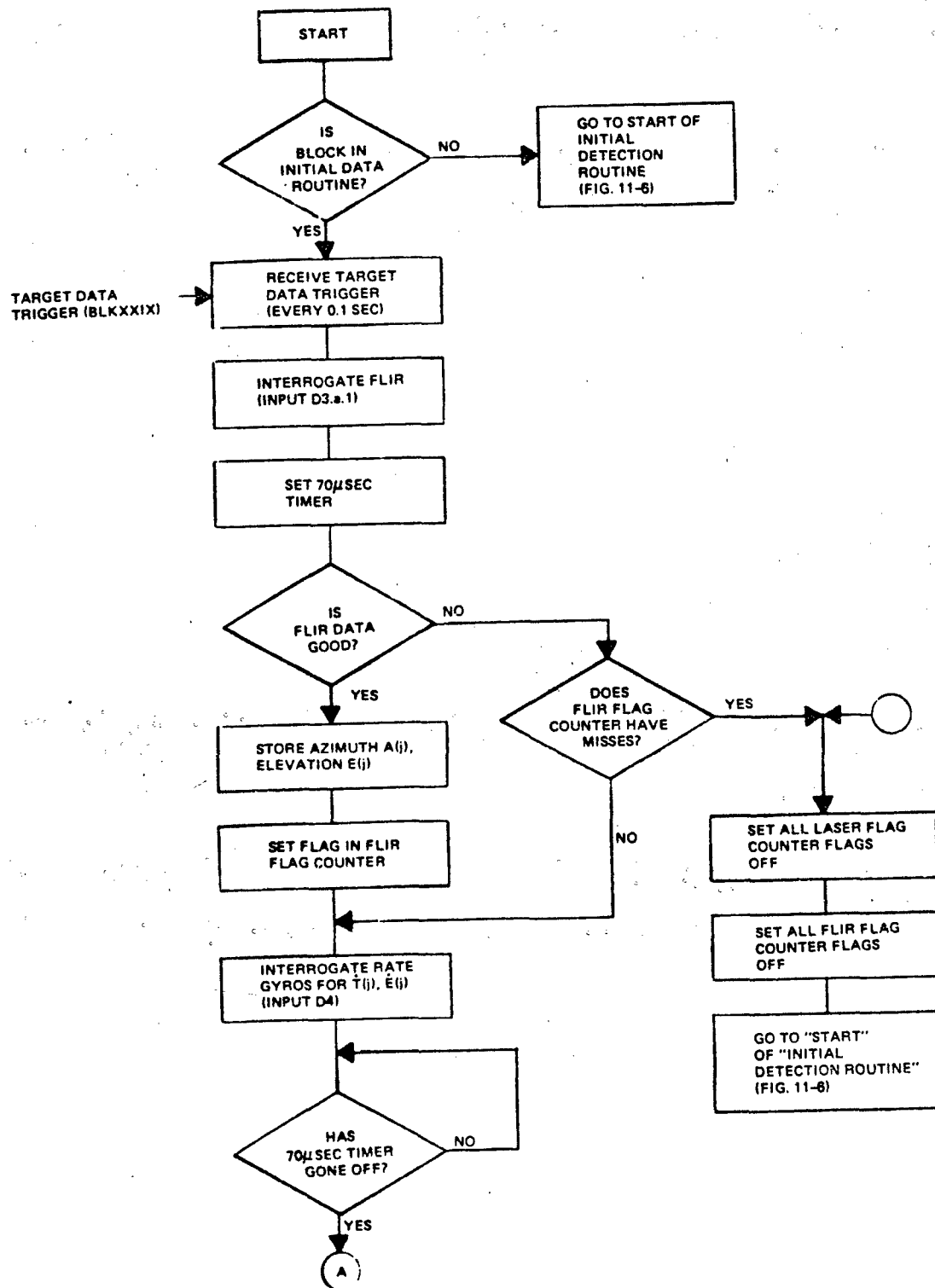


Figure 11-7. Initial Data Routine (Sheet 1 of 2).

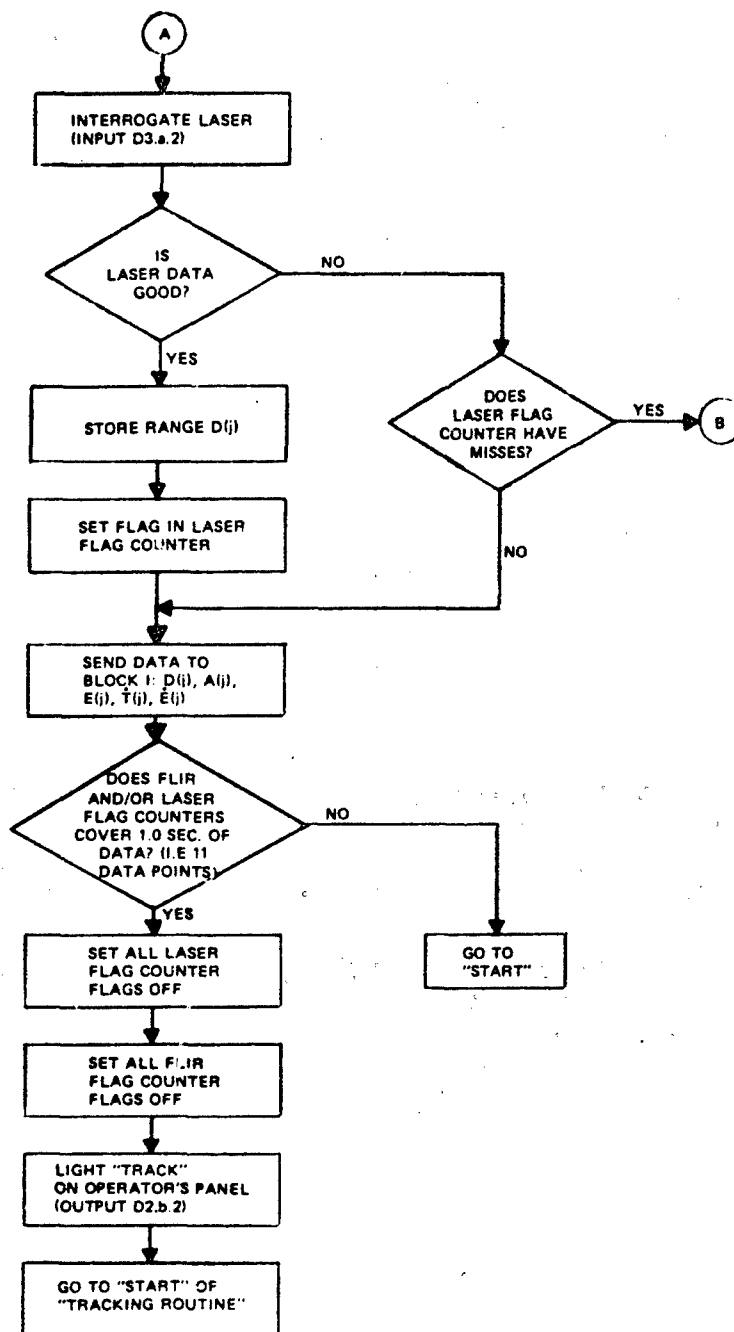


Figure 11-7. Initial Data Routine (Sheet 2 of 2).

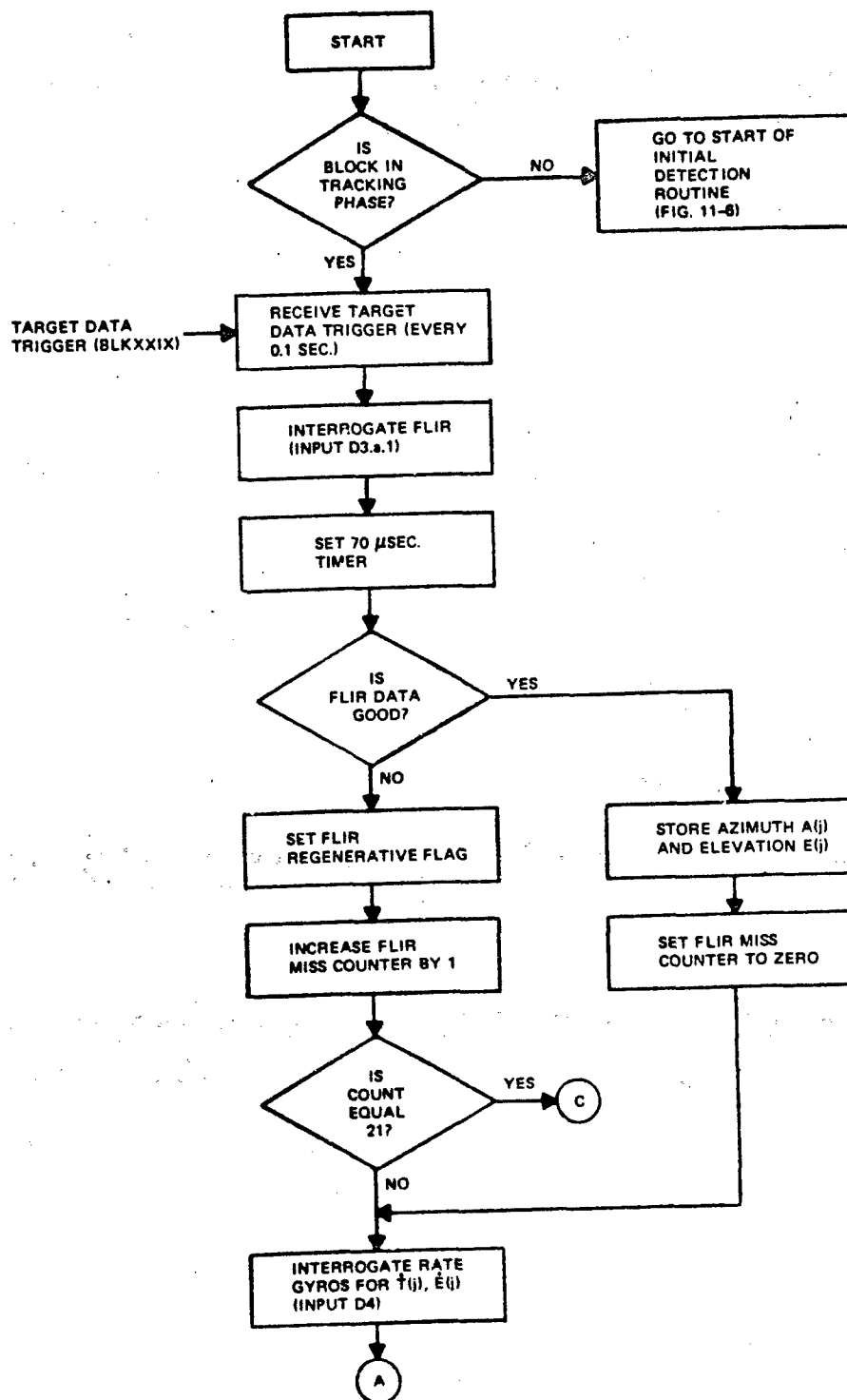


Figure 11-8. Tracking Routine (Sheet 1 of 3)

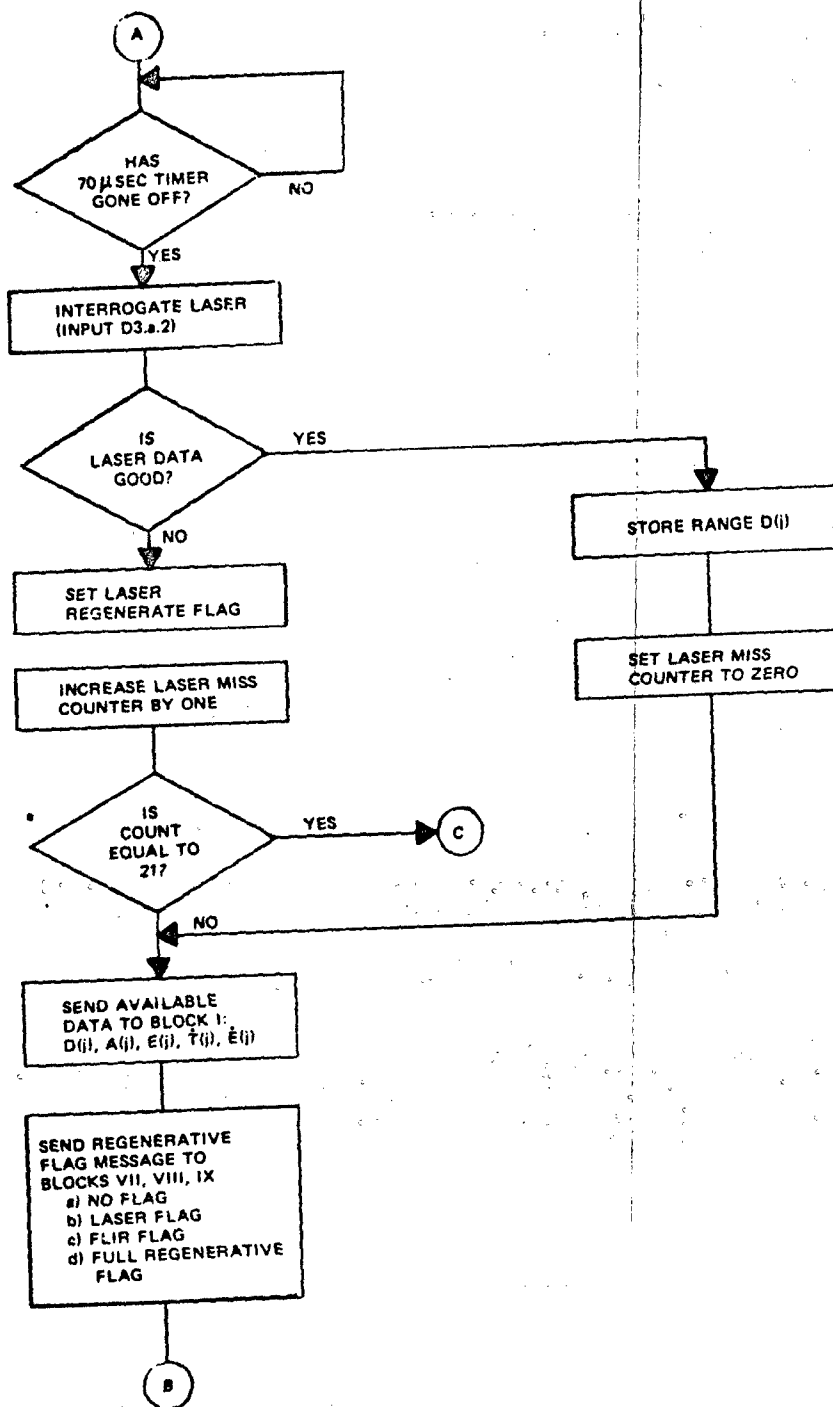


Figure 11-8. Tracking Routine (Sheet 2 of 3).

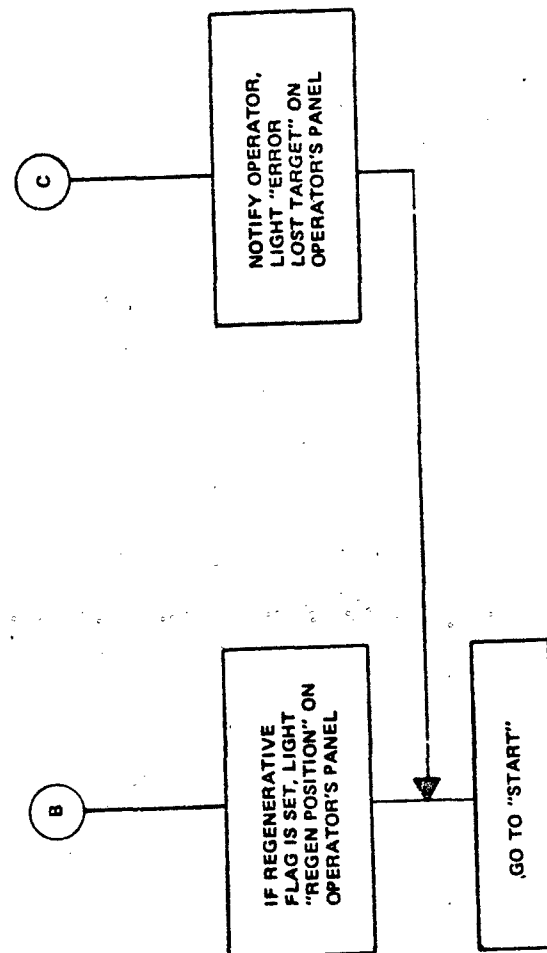


Figure 11-8. Tracking Routine (Sheet 3 of 3).

- a) Regeneration is required in range and/or angle due to sensor failures, target behind an obstacle, etc.
- b) Operator should be notified when 2.0 seconds of missed data have elapsed.

6. Outputs:

6.1 Target Data: To Block I.

$D(j)$ ,  $A(j)$ ,  $E(j)$ ,  $\dot{T}(j)$ , and  $\dot{E}(j)$  exactly equal to the received data.

6.2 Regeneration Triggers:

- a) Full Regenerative Trigger: Sent to Blocks VII, VIII, and XI. Indicates that neither FLIR nor laser data were received this scan interval.
- b) Laser Regenerative Trigger: Sent to Blocks VII and XI (not VIII). Indicates that no laser data were received. Regenerative tracking rates to the sensor are not required in this case.
- c) FLIR Regenerative Trigger: Sent to Blocks VII, VIII, and XI. Indicates that no FLIR data were received. Difference between this case and that involving the Full Regenerative Trigger (Item 6.2) is type of data insertion by Block VII.

7. Comments: None

Block VII: Regenerative Data Insertion for Missed Data

- 1. Function: To insert regenerative data in range, azimuth, and/or elevation for the missed range, azimuth, and/or elevation data from the sensors at the input to the target tracking function (Blocks I and II).
- 2. Frequency: Whenever a regenerative trigger is received, no more frequent than every 0.1 second.
- 3. Inputs:
  - 3.1 Regenerative Triggers: From Block VI.
    - a) Full Regenerative Trigger: Indicates neither laser range data nor FLIR azimuth and elevation data were received.

- b) Laser Regenerative Trigger: Indicates only laser range data not received from the sensor head. Hence, only regenerative range must be supplied.
- c) FLIR Regenerative Trigger: Indicates only FLIR azimuth and elevation data were not received from the sensor head and hence must be supplied from regeneration.

### 3.2 Regenerated Position Data: From Block XI.

$D(r)$  = Regenerated target range at time  $j$ .

Magnitude: 100m to 10,000m

Accuracy: \_\_\_\_\_ m.

$A(r)$  = Regenerated target azimuth at time  $j$ , positive clockwise from front of vehicle.

Magnitude: 0 deg to 360 deg.

Accuracy: \_\_\_\_\_ mil.

$E(r)$  = Regenerated target elevation at time  $j$ , positive up.

Magnitude: -5 deg to +85 deg.

Accuracy: \_\_\_\_\_ mil.

### 4. Constants: None

### 5. Operations Performed:

- a) If Full Regenerative Trigger is received, send  $D(r)$ ,  $A(r)$ , and  $E(r)$  to Block I to replace missing sensor values for these parameters.
- b) If Laser Regenerative Trigger is received, send  $D(r)$  only to Block I to replace missing laser sensor range data.
- c) If FLIR Regenerative Trigger is received, send  $A(r)$  and  $E(r)$  only to Block I to replace missing FLIR sensor angle data.

### 6. Outputs:

#### 6.1 Regenerative Target Position Data: To Block I.

As determined by the operational logic, send  $D(r)$ ,  $A(r)$  and/or  $E(r)$ . Magnitudes and accuracies are same as Item 3, Inputs.

### 7. Comments: None

### Block VIII: Regenerative Target Velocity

1. Function: In the Semi-Automatic Fire Control Mode, to compute regenerative transverse and elevation rates for ordering the sensor servos during tracking periods when no target azimuth and elevation angular data is received from the FLIR. These regenerative rates are basically one sample period predictions of the target's position based upon the previously generated smoothed target position and velocity data. By use of these rates, the sensors will follow the target and, hopefully, be able to automatically reacquire it.

In the Test Mode, these signals are continuously generated so the tracking sensors will follow the simulated target.

As a growth feature, regenerative signals can be used to prevent the sensor servos from "getting behind" during periods of anticipated excessive target angular rates.

#### 2. Frequency:

2.1 Semi-Automatic Fire Control Mode: Whenever FLIR azimuth and elevation data are missing, maximum frequency of 10 times per second.

2.2 Test Mode: Every 0.1 second

#### 3. Inputs:

3.1 Current Smoothed Target Position and Velocity: From Block II

$X(j)$  = Distance to target from vehicle at time  $t(j)$ , positive to right  
Magnitude: -10,000m to 10,000m.  
Accuracy: \_\_\_\_\_ m.

$\dot{X}(j)$  = Velocity of target at  $t(j)$ , positive to right  
Magnitude: -400m/sec to +400m/sec.  
Accuracy: \_\_\_\_\_ m/sec.

$Y(j)$  = Distance to target from vehicle at  $t(j)$ , positive forward  
Magnitude: -10,000m to 10,000m.  
Accuracy: \_\_\_\_\_ m/sec.

$\dot{Y}(j)$  = Velocity of target at time  $t(j)$ , positive forward  
Magnitude: -400m/sec to +400m/sec.

$\underline{Z}(j)$  = Target altitude measured from vehicle at time  $t(j)$ , positive up.

Magnitude: -900m to +8,000m

Accuracy: \_\_\_\_\_ m

$\dot{\underline{Z}}(j)$  = Elevation rate measured from vehicle at time  $t(j)$ , positive up.

Magnitude: -400m/sec to +400m/sec.

Accuracy: \_\_\_\_\_ m/sec.

3.2 Regeneration Trigger: From Block VI

Either the Full Regenerative Trigger or the FLIR Regenerative Tripper.

4. Constants: None

5. Operations Performed:

a. Computation of current ground range,  $R(j)$

$$R(j) = \text{square root of } [\underline{X}(j)^2 + \underline{Y}(j)^2]^{1/2}$$

b. Computation of current slant range,  $D(j)$

$$D(j)^2 = R(j)^2 + \underline{Z}(j)^2$$

c. Computation of regenerative elevation rate,  $\dot{E}(r)$

$$\dot{E}(r) = -[\dot{\underline{X}}(j) \underline{X}(j) + \dot{\underline{Y}}(j) \underline{Y}(j)]\underline{Z}(j)/[D(j)^2 R(j)] + \dot{\underline{Z}}(j)R(j)/D(j)^2$$

d. Computation of regenerative transverse rate,  $\dot{T}(r)$

$$\dot{T}(r) = [\dot{\underline{X}}(j) \underline{Y}(j) - \dot{\underline{Y}}(j) \underline{X}(j)]/[D(j)R(j)]$$

6. Outputs: To sensor servos, peripheral device D6, the two rates,  $\dot{E}(r)$  and  $\dot{T}(r)$ .

$\dot{E}(r)$  = Elevation rate regenerated, positive up

Magnitude: -25 deg/sec to +25 deg/sec.

Accuracy: \_\_\_\_\_ deg/sec.

$\dot{T}(r)$  = Transverse rate regenerated, positive right

Magnitude: -90 deg/sec to +90 deg/sec.

Accuracy: \_\_\_\_\_ deg/sec.

7. Comments: Any value used in these equations besides  $D(j)$  and  $R(j)$  may be close to zero. All division is done with  $D(j)$  and  $R(j)$  but some intermediate products may be small.

Block IX: Muzzle Velocity (from Operator)

1. Function: To provide muzzle velocity for ballistic corrections, as inputted by the Operator during Initialization II Mode. Entered value is continuously displayed on the Operator's Panel.

2. Frequency:

2.1 Operator Panel Polling: Operator's Panel is polled for a change in the entered muzzle velocity every 0.1 second during Initialization II Mode.

2.2 Ballistic Corrections: Value of muzzle velocity to be available for ballistic corrections during Initialization II and for every ballistic computation update, i. e., every 0.1 second, during the Real Time Semi-Automatic Fire Control Mode.

3. Inputs:

3.1 Polling Trigger: From Block XXIX, Real Time Clock, polling trigger is received every 0.1 second for polling the Operator's Panel output buffer (peripheral device D7).

3.2 Operator's Panel: From peripheral device D7, the value of the muzzle velocity: Either

(i) PRESET value trigger

(ii) Entered value of muzzle velocity,  $V(o)$ ,  
Magnitude: 930m/sec to 1570m/sec.  
Accuracy: 10m/sec increments

3.3 Projectile Type: From Block XXXIV, the type of projectile to be fired (one of five).

4. Constants:

4.1 Preset Value: Value to be used if Operator enters PRESET.

- a. If Projectile Type is "Subcaliber," velocity magnitude is 1500m/sec.
- b. If Projectile Type is not "Subcaliber," velocity magnitude is 1200 m/sec. (May be expanded to separate preset values for each projectile type).

4.2 Stored Value: Last value entered by the Operator.

V(o) ■ Muzzle velocity entered by the Operator.

Magnitude: 930m/sec to 1570m/sec.

Increments: 10m/sec.

5. Operations Performed:

- a. Upon receipt of the Polling Trigger (Item 3.1), projectile type buffer store of Block XXXIV is polled for any change:
  - (i) If no change, go to Step d.
  - (ii) If changed, go to Step b.
- b. Is the change to or from a subcaliber round?
  - (i) If no change, go to Step d. (There are four standard caliber rounds, so a change between one of these four results in this logical answer).
  - (ii) If changed, to to Step c.
- c.
  - (i) If change is to a subcaliber round, enter 1500m/sec from Preset Value memory location (Item 4.1) into Stored Value memory location (Item 4.2) and go to Step g.
  - (ii) If change is from a subcaliber round, enter 1200m/sec from Preset Value memory location into Stored Value memory location and go to Step g.
- d. Poll the Operator's Panel output buffer for a change in the muzzle velocity (part of peripheral device D7).
  - (i) If no change, end processing
  - (ii) If changed, to to Step e if change is to PRESET or to Step f if change is a new value.
- e. If change is PRESET and if
  - (i) Projectile type is subcaliber, enter 1500m/sec in Stored Value memory location and to Step g.
  - (ii) Projectile type is not subcaliber, enter 1200m/sec in Stored Value memory location and go to Step g.
- f. If change is a new value, enter this value in Stored Value memory location and go to Step g.

g. Display new value in LED readout for muzzle velocity.

h. Stored Value of muzzle velocity available for ballistic corrections during both Initialization II Mode and Real Time Fire Control Mode.

6. Outputs:

6.1 Ballistic Corrections: To Block IV during Initialization II Mode, and to Operator's Panel LED display during all modes:

$V(o)$  = Muzzle velocity. Magnitude and accuracy as given in Item 4.2 above.

7. Comment: None

Block X: Is Miss Distance Data Received?

1. Function: To determine whether miss distance is available on the tracking sensor output buffer, and if available to input the data to the computer for processing.

2. Frequency: Once every 0.1 second

3. Inputs:

3.1 Tracking Sensor Output Buffer: From peripheral device D3.b, digital measured projectile miss distance and measurement time.

$\Delta T(mk)$  = Measured transverse projectile miss angle at time  $t(k)$ ; positive to the right. Value consists of mean miss angle of all projectiles within the field of view.  
Magnitude: -4 deg to +4 deg except -0.008 deg (0.14 mil) to +0.008 deg cannot be resolved.

Accuracy: 0.25 mil.

$\Delta E(mk)$  = Measured elevation projectile miss angle at time  $t(k)$ , positive up. Value consists of mean miss angle of all projectiles within the field of view.

Magnitude: -2 deg to +2 deg except - 0.008 deg to +0.008 deg cannot be resolved.

Accuracy: 0.25 mil.

$\Delta D(mk)$  = Measured projectile-to-target range. Value consists of mean of all projectiles within the range gate. Positive before projectile reaches the target's range.

Magnitude: -600m to +600m

Accuracy: 0.25m

$t(k)$  = Time of day when measurement was made.

Magnitude: 0 to about 5 min should be adequate range.

Accuracy: 0.005 sec.

3.2 Miss Data Trigger: From Block XXIX, sampling trigger for polling the sensors for projectile miss data, occurs at 10 times per second with a 0.005 second timing accuracy.

4. Constants: None.

5. Operations Performed:

5.1 Interrogate Tracking Sensor Output Buffer: Upon receipt of the Miss Data Trigger (every 0.1 second), interrogate the Tracking Sensor Output Buffer (peripheral device D3.b) to determine if miss distance measurement data are present.

5.2 Input Miss Distance Data: If projectile miss distance data are present on buffer, read data into computer and initiate Block XXIV processing.

5.3 No Miss Distance Data: If no projectile miss distance data are present, generate No Miss Distance Computation Trigger and transmit to Block XVII for processing.

6. Outputs:

6.1 If Miss Distance Data Present: To Block XXIV.

$\Delta T(mk)$  = Measured transverse mean projectile miss angle.  
(Identical to input signal, Item 3.1)

$\Delta E(mk)$  = Measured elevation mean projectile miss angle  
(Identical to input signal).

$\Delta D(mk)$  = Measured range of mean projectile miss.  
(Identical to input signal).

$t(k)$  = Time of day when measurement was made.  
(Identical to input signal).

6.2 If Miss Distance Data Absent: To Block XVII, send No Miss Distance Computation Trigger to initiate Block XVII function.

7. Comments: None

Block XI: Regenerative Target Position

1. Function: To compute regenerative, or one-scan-extrapolated, target position data based upon the current best estimate of the smoothed target position. These regenerated values are used to replace missing laser and/or FLIR target coordinates and to position the laser range gate (range servo). Such replacement is necessary (a) since the FLIR/laser sensor provides polar coordinate target data but all tracking is done in Cartesian coordinates and (b) the postulated laser sensor does not have a velocity servo loop.

2. Frequency: Whenever a regenerative trigger is received, but no more frequently than once per scan or every 0.1 second.

3. Inputs:

3.1 Regenerative Triggers: From Block VI.

a) Full Regenerative Trigger: Indicates neither laser range data nor FLIR azimuth and elevation data were received.

b) Laser Regenerative Trigger: Indicates that no laser range data were received from the sensor.

c) FLIR Regenerative Trigger: Indicates that no FLIR azimuth and elevation data were received from the sensor head.

3.2 Smoothed Target Data: From Block II.

$X(j-1)$  = Smoothed cross position of the target at time interval  $t(j-1)$ , right of vehicle is positive.

Magnitude: -10,000m to +10,000m.

Accuracy: \_\_\_\_\_ m.

$Y(j-1)$  = Smoothed lengthwise position of the target at time interval  $t(j-1)$ , positive forward.

Magnitude: -10,000m to +10,000m.

Accuracy: \_\_\_\_\_ m.

$\underline{Z}(j-1)$  = Smoothed vertical position of the target at time interval  $t(j-1)$ , positive up.

Magnitude: -900m to +8,000m.

Accuracy: \_\_\_\_\_ m.

$\dot{\underline{X}}(j-1)$  = Smoothed cross velocity of the target at time interval  $t(j-1)$ , positive right.

Magnitude -400m/sec to +400m/sec.

$\dot{\underline{Y}}(j-1)$  = Smoothed lengthwise velocity of the target at time interval  $t(j-1)$ , positive forward.

Magnitude: -400m/sec to +400m/sec.

$\dot{\underline{Z}}(j-1)$  = Smoothed vertical velocity of the target at time interval  $t(j-1)$ , positive up.

Magnitude: -400m/sec to +400m/sec.

Accuracy: \_\_\_\_\_ m/sec.

4. Constants: None

5. Operations Performed:

5.1 Initialization: The operations of this function are only performed if one of the three regenerative triggers of Section 3.1 are received.

5.2 Target Position Extrapolation: To extrapolate forward by one scan the current smoothed target position, i. e., to obtain the regenerative target position in Cartesian coordinates at time interval  $t(j)$  based upon the smoothed position at time interval  $t(j-1)$ . These operations are performed whenever a regenerative trigger is received.

$$X(r) = \underline{X}(j-1) + \dot{\underline{X}}(j-1) \Delta t$$

$$Y(r) = \underline{Y}(j-1) + \dot{\underline{Y}}(j-1) \Delta t$$

$$Z(r) = \underline{Z}(j-1) + \dot{\underline{Z}}(j-1) \Delta t$$

where  $\Delta t = 0.1$  sec = Period between sensor data inputs, i. e., between time intervals  $t(j-1)$  and  $t(j)$ .

5.3 Regenerative Target Position: To convert the regenerative target position in Cartesian coordinates to that in polar coordinates:

- a) Performed for all three regenerative triggers.

$$D(r) = [X(r)^2 + Y(r)^2 + Z(r)^2]^{1/2}$$

- b) Performed for the Full Regenerative Trigger and FLIR Regenerative Trigger only.

$$E(r) = \sin^{-1}[A(r)/D(r)]$$

$$A(r) = \begin{cases} \sin^{-1}[X(r)/D(r) \cos E(r)] \\ \text{or} \\ \cos^{-1}[Y(r)/D(r) \cos E(r)] \end{cases}$$

6. Outputs:

- 6.1 Regenerative Target Range: To Block VII and the laser range sensor servo (peripheral device D6)

$D(r)$  = Regenerated target range at time  $t(j)$ , positive out.

Magnitude: 100m to 10,000m.

Accuracy: \_\_\_\_\_ m

- 6.2 Regenerative Target Azimuth and Elevation: To Block VII

$A(r)$  = Regenerated target azimuth at time  $t(j)$ , measured clockwise from front of vehicle.

Magnitude: 0 deg to 360 deg.

Accuracy: \_\_\_\_\_ mil.

$E(r)$  = Regenerated target azimuth at time interval  $t(j)$ , positive up.

Magnitude: -5 deg to +85 deg.

Accuracy: \_\_\_\_\_ mil.

7. Comments: None

Block XII: Ballistic Computation to Reach Current Target Position

1. Function: To compute the projectile time of flight that would have been required to hit the target at its current position, i. e., to hit the target at current time  $t(j)$ .

2. Frequency: When a valid projectile miss distance measurement is made, maximum rate of 10 times per second.

3. Inputs:

3.1 Current Smoothed Target Position in Polar Coordinates: From Block XX.

$\underline{D}(mj)$  = Slant range from AFAADS gun to current target position, i. e., position at time  $t(j)$ , positive out.

Magnitude: 100m to 10,000m

Accuracy: \_\_\_\_\_ m.

3.2 Valid Miss Distance Trigger: From Block XXVII, a Trigger indicating that the ballistic miss distance computations should be carried out.

4. Constants: See Item 5 below.

5. Operations Performed: Upon receipt of the Valid Miss Distance Trigger, compute the projectile time of flight using the same ballistic equation used in Block IV, Item 5.2, Step e; namely,

$$t(pj) = a \underline{D}(mj) + b \underline{D}^2(mj) + c \underline{D}^3(mj) + d \underline{D}^4(mj) + e \underline{D}^5(mj)$$

where:  $t(pj)$  = Projectile time of flight required to hit a target at its current range  $\underline{D}(mj)$

$\underline{D}(mj)$  = Current target range, Item 3.1 above.

a, b, c, d, e = Ballistic coefficients computed in Block IV, Item 4.4.1 during the Initialization Mode.

This computation differs from that in Block IV since a definitive problem exists; namely, to calculate the flight time over a specified range. Block IV requires an iterative solution with Block III, in order to match the predicted target position against the predicted projectile position for impact.

The output of this block is the projectile time of flight,  $t(pj)$  required to hit a target at range  $\underline{D}(mj)$ .

6. Outputs:

6.1 Time of Flight: To Block XXXIX.

$t(poj)$  = Predicted projectile time of flight to hit the target at its current position.

Magnitude: 0 sec to 10 sec.

Accuracy: 0.005 sec (see Item 7, Comment)

7. Comments: Accuracy of ballistic computation and hence of  $t(poj)$  may not have to be as great as in Block IV. Accuracy requirement is dictated by the matching requirements of Block XIV.

Block XIII: Ballistic Data Store

1. Function: To provide storage for the orders given to the gun mount during at least one projectile time of flight. This data is employed in the projectile miss distance computations to subtract out the miss due to target maneuvers (Block XXVIII) and in the bias correction algorithm (Block XVI).

2. Frequency:

2.1 Data Storage Update or Input: Every 0.1 second.

2.2 Memory Interrogation or Output: Whenever a valid projectile miss distance measurement is made (at most every 0.1 second), a series of memory interrogations is initiated.

3. Inputs:

3.1 Data Storage Update or Input: Whenever Block XXX, the Ballistic Data Input Register, is full, the following data are inputted to the memory:

$t(1)$  = Time of day for the data, time interval 1.

Magnitude: Clock time, 5 min. maximum should be adequate.

Accuracy: 0.005 sec.

$t(pl)$  = Projectile time of flight for projectile fired at time interval 1.

Magnitude: 0 sec to 10 sec.

Accuracy: 0.005 sec.

$A(pl)$  = Predicted target azimuth,  $t(pl)$  seconds from time interval 1; measured clockwise from front of vehicle.

Magnitude: 0 deg to 360 deg.

Accuracy: \_\_\_\_\_ mils.

$E(p1)$  = Predicted target elevation,  $t(p1)$  seconds from time interval 1, positive up.

Magnitude: -5 deg to +85 deg.

Accuracy: \_\_\_\_\_ mils.

$\delta(1)$  = Corrected gun lead angle in azimuth; positive clockwise.

Magnitude: -90 deg. to +90 deg.

Accuracy: 0.25 mils.

$\sigma(1)$  = Corrected gun lead angle in elevation, positive up.

Magnitude: -30 deg. to +70 deg.

Accuracy: 0.25 mils.

### 3.2 Memory Interrogations:

3.2.1 Fire Time Inquiry: From Block XIV, inquiry for the clock times of projectile fire entries. These times are used in the matching process of Block XIV.

3.2.2 Request Target Azimuth and Elevation: From Block XV, table indices corresponding to times  $t(n)$  and  $t(n + 1)$  with request for predicted target azimuth and elevation angles,  $A(p,n)$ ,  $E(p,n)$ ,  $A(p,n + 1)$ , and  $E(p,n + 1)$ . These times  $t(n)$  and  $t(n + 1)$  lie on either side of the equal range time.

3.2.3 Request Target Lead Angles: From Block XXI, table indices corresponding to times  $t(n)$  and  $t(n + 1)$  with request for gun lead angles in azimuth and elevation,  $\delta(p,n)$ ,  $\sigma(p,n)$ ,  $\delta(p,n + 1)$  and  $\sigma(p,n + 1)$ .

3.2.4 Request Projectile Time of Flight: From Block XVI, table indices corresponding to times  $t(n)$  and  $t(n + 1)$  with request for predicted projectile time of flight,  $t(p,n)$  and  $t(p,n + 1)$ .

### 4. Constants:

4.1 Ballistic Data Store: The file shall contain about 100 entries of ballistic data storage, i. e., 10 seconds worth of data. Each entry consists of the six quantities listed under Item 3.1.

### 5. Operations Performed:

5.1 Data Storage Update: Replace the oldest data in storage, Item 4.1, with the new data described in Item 3.1.

### 5.2 Memory Interrogation:

5.2.1 Matching Fire Times: In response to the time of projectile fire input (Item 3.2.1 above), provide the time of day and file indices from the Ballistic Data Store (Item 4.1) that lie the closest on either side of the input time of fire signal.

5.2.2 Data Retrieval: In response to the indice requests of Items 3.2.2, 3.2.3, and 3.2.4, supply the requested data from the Ballistic Data Store (Item 4.1).

6. Outputs:

6.1 Stored Time of Projectile Fire: To Block XIV

$t(1)$  = Each of the clock times of projectile fire that are in the Ballistic Data Store (Item 4.1) in response to request to match.

Magnitude: Clock time, 5 min. maximum

Accuracy: 0.005 sec.

6.2 Matched Fire Time Indices: To Block XIV, the indices of the files corresponding to clock times  $t(n)$  and  $t(n + 1)$ , where  $t(n)$  and  $t(n + 1)$  are the selected adjacent times to the time of projectile fire,  $t(j-p)$ , in Block XIV.

6.3 Requested Target Azimuth and Elevation: To Block XV.

$A(p,n)$  and  $A(p,n + 1)$  = Predicted target azimuth angles for projectiles fired at times  $t(n)$  and  $t(n + 1)$ , measured clockwise.

Magnitude: 0 to 360 deg.

Accuracy: \_\_\_\_\_ mils.

$E(p,n)$  and  $E(p,n + 1)$  = Predicted target elevation angles for projectiles fired at times  $t(n)$  and  $t(n + 1)$ , positive up.

Magnitude: -5 deg. to +85 deg.

Accuracy: \_\_\_\_\_ mils.

6.4 Requested Target Lead Angles: To Block XXI.

$\delta(p,n)$  and  $\delta(p,n + 1)$  = Predicted gun lead angles in azimuth for projectiles fired at times  $t(n)$  and  $t(n + 1)$ , positive clockwise.

Magnitude: -90 deg. to +90 deg.

Accuracy: 0.25 mil.

$\sigma(p,n)$  and  $\sigma(p,n + 1)$  = Predicted gun lead angles in elevation for projectiles fired at times  $t(n)$  and  $t(n + 1)$ ; positive up.

Magnitude: -30 deg. to +70 deg.

Accuracy: 0.25 mil

6.5 Requested Projectile Time of Flight: To Block XVI

$t(p,n)$  and  $t(p,n + 1)$  = Projectile time of flight for targets fired at  $t(n)$  and  $t(n + 1)$ .

Magnitude: 0 sec. to 10 sec.

Accuracy: 0.005 sec.

7. Comments: None.

Block XIV: Match Fire Times

1. Function:

a) To determine which of the two entries in the Ballistic Data Store (Block XIII, Item 4.1) correspond the closest in time to the computed time of projectile fire,  $t(j-p)$ .

b) To compute the fractional correction to be applied to the stored parameters for linear interpolation to time of projectile fire,  $t(j-p)$ .

2. Frequency: Whenever a valid projectile miss distance measurement is made, no more often than 10 times per second.

3. Inputs:

3.1 Computed Time of Projectile Fire: From Block XXXIX.

$t(j-p)$  = Time of projectile fire for a projectile to hit the target at its current position.

Magnitude: Clock time, 5 min. maximum

Accuracy: 0.005 sec.

3.2 Stored Times of Projectile Fire: From Block XIII

$t(1)$  = Each of the clock times of projectile fire that are in the Ballistic Data Store (Block XIII, Item 4.1).

Magnitude: Clock time, 5 min. maximum

Accuracy: 0.005 sec.

3.3 Matched Fire Time Indices: From Block XIII, the indices for the Ballistic Data Store table entries corresponding to clock times of projectile fire  $t(n)$  and  $t(n + 1)$ , where these clock times are the adjacent times in the table on either side of the time of projectile fire  $t(j-p)$ ,

4. Constants: None

5. Operations Performed: The logical operations are:

- a) Time Match: Match  $t(j-p)$  against the various clock times of projectile fire,  $t(1)$ , in the Ballistic Data Store (Block XIII, Item 4.1) to determine those  $t(1)$ 's which lie adjacent to and on either side of  $t(j-p)$ . That is find the closest  $t(n)$  and  $t(n + 1)$  such that

$$t(n) \leq t(j-p) \leq t(n + 1)$$

The times  $t(n)$  and  $t(n + 1)$  are the selected times for the remainder of the processing of this block.

- b) Index Retrieval: For times  $t(n)$  and  $t(n + 1)$ , extract from the Ballistic Data Store the corresponding file indices.
- c) Corrective Factor: Compute the correction factor  $C(j)$  to be applied to the various data items in the Ballistic Data Store to linearly interpolate them to correspond to the time of projectile fire  $t(j-p)$

$$C(j) = \frac{t(j-p) - t(n)}{t(n + 1) - t(n)}$$

6. Outputs:

6.1 Match Time Data: To Blocks XV, XVI, and XXI.

1. Indices of table entries for match times  $t(n)$  and  $t(n + 1)$
2.  $C(j)$  - Corrective factor for equal range point correction  
Magnitude: 0 to 1.0, a numeric  
Accuracy: 0.05

6.2 Inquiry: To Block XIII

Inquiry for clock time entries in the Ballistic Data Store table (Block XIII, Item 4.1) as a part of the matching process (Item 5, step a).

7. Comments: A trade-off exists between memory in Block XIII and data processing in Blocks XIV, XV, XVI, and XXI.

Method Used: Values are entered in Block XIII, Ballistic Data Store, every 0.1 sec such that a linear interpolation between values is required to obtain the equal range values for the needed parameters.

Alternate Method: Additional values are stored in Block XIII, such that the closest time match provides sufficiently accurate values of the parameters that the corrective factor  $C(j)$  need not be used.

Block XV: Extract Predicted Angles for Equal Time Points

1. Function: To extract from Block XIII, Ballistic Data Store, the required data and then compute the predicted target azimuth and elevation for the equal range point between the target and projectile.

2. Frequency: Whenever a valid projectile miss distance measurement is made, no more often than every 0.1 second.

3. Inputs:

3.1 Match Time Data: From Block XIV

1. Indices of table entries in Block XIII that provide the best time matches with time of projectile fire  $t(j-p)$ , i. e., indices for times  $t(n)$  and  $t(n + 1)$  in the Ballistic Data Store (Block XIII, Item 4.2).

2.  $C(j)$  = Corrective factor for equal range point correction.

Magnitude: 0 to 1.0, a numeric

Accuracy: 0.05

3.2 Target Angles: From Block XIII

$A(p,n), A(p,n + 1)$  = Predicted target azimuth for projectiles fired at times  $t(n)$  and  $t(n + 1)$ ; positive clockwise.

Magnitude: 0 to 360 deg.

Accuracy: \_\_\_\_\_ mils

$E(p,n), E(p, n + 1)$  = Predicted target elevation for projectiles fired at times  $t(n)$  and  $t(n + 1)$ ; positive up.

Magnitude: -5 to +85 deg.

Accuracy: \_\_\_\_\_ mils.

4. Constants: None

5. Operations Performed:

5.1 Extract: Using the table indices for times  $t(n)$  and  $t(n + 1)$ , retrieve from the Block XIII, the corresponding values of  $A(p,n)$ ,  $E(p,n)$ ,  $A(p,n + 1)$ ,  $E(p, n + 1)$ .

5.2 Interpolate: Determine by linear interpolation the equal range values of predicted target values:

$$A(p,j,t(j-p)) = C(j) [A(p,n + 1) - A(p,n)] + A(p,n)$$

$$E(p,j,t(j-p)) = C(j) [E(p,n + 1) - E(p,n)] + E(p,n)$$

6. Outputs:

6.1 Request: To Block XIII

Table indices for entries corresponding to  $t(n)$  and  $t(n + 1)$  and request for azimuth and elevation data.

6.2 Predicted Target Values: To Block XXVIII

$A(p,j,t(j-p))$  = Predicted target azimuth angle at the equal range point; positive clockwise.

Magnitude: 0 to 360 deg.

Accuracy: \_\_\_\_\_ mils

$E(p,j,t(j-p))$  = Predicted target elevation angle at the equal range point; positive up.

Magnitude: -5 deg to +85 deg.

Accuracy: \_\_\_\_\_ mils.

7. Comments: None

#### Block XVI: Bias Correction Algorithm

1. Function: To allocate the residual projectile miss errors into bias corrections on muzzle velocity, gun azimuth (lead angle), and gun elevation (lead angle). A sample calculation is given in Appendix E.

2. Frequency: Whenever valid projectile miss distance measurement has been made, no more frequent than every 0.1 second.

Block XVI: Bias Correction Algorithm

1. Function: To allocate the residual projectile miss errors into bias corrections on muzzle velocity, gun azimuth (lead angle), and gun elevation (lead angle). A sample calculation is given in Appendix E.
2. Frequency: Whenever valid projectile miss distance measurements have been made, no more frequent than every 0.1 second.

3. Inputs:

3.1 Residual Miss Distance Measurement: From Block XXXVIII

$\Delta T(j)$  = Residual transverse miss angle in the horizontal plane between target and projectile when both are at the same range. Positive if projectile is right of predicted target position.

Magnitude: -2 to +2 deg.

Accuracy: 0.5 mil.

$\Delta E(j)$  = Residual elevation miss angle in vertical plane between target and projectile. Positive if projectile is above predicted target position.

Magnitude: -2 to +2 deg.

Accuracy: 0.5 mil.

3.2 Target Elevation: From Block XX

$E(mj)$  = Current smoothed measured target elevation angle, positive up.

Magnitude: -5 to +85 deg.

Accuracy: \_\_\_\_\_ mil.

3.3 Gun Lead Angles: From Block XXI

$L(j)$  = Total lead angle for projectile arriving at target at time  $t(j)$ .

Magnitude: 0 deg to +90 deg.

Accuracy: 0.25 mil

$\delta(pj)$  = Azimuth lead angle for projectile arriving at target at time  $t(j)$ , positive clockwise.

Magnitude: -90 deg to +90 deg.

Accuracy: 0.25 mil

$\sigma(pj)$  = Elevation lead angle for projectile arriving at target at time  $t(j)$ , positive up.  
Magnitude: -30 to +70 deg.  
Accuracy: 0.25 mil.

3.4 Time of Flight: From Block XIII

$t(p,n), t(p,n + 1)$  = Predicted projectile time of flight for projectiles fired at clock times  $t(n)$  and  $t(n + 1)$ .  
Magnitude: 0 sec to 10 sec.  
Accuracy: 0.005 sec.

3.5 Match Time Data: From Block XIV

1. Indices of table entries in Block XIII that provide the best time matches with time of projectile fire  $t(j-p)$ ; i. e., indices for times  $t(n)$  and  $t(n + 1)$ .
2.  $C(j)$  = Corrective factor for equal range point correction.  
Magnitude: 0 to 1.0, a numeric  
Accuracy: 0.05

4. Constants:

4.1 Bias Variances:

$\sigma^2(A)$  = Azimuth bias variance

Magnitude: less than  $(10 \text{ mils})^2$

Accuracy: 0.1%

$\sigma^2(E)$  = Elevation bias variance

Magnitude: less than  $(10 \text{ mils})^2$ , and equal to the azimuth bias variance.

Accuracy: 0.1%

$\sigma^2(V)$  = Muzzle velocity bias variance expressed in terms of an equivalent angular error.

Magnitude:  $(8 \text{ mils})^2$

Accuracy: 0.1%

Note: The equivalent angular muzzle velocity error, whose variance is defined above, is the following function of the actual muzzle velocity error:

$$a(vj) = \left[ \frac{\Delta V(pj)}{V(p)} \right] L(j)$$

where:  $a(vj)$  = Equivalent angular muzzle velocity error at time  $t(j)$ .

$\Delta V(pj)$  = Muzzle velocity error in meters per second at time  $t(j)$ .

$V(a)$  = Average projectile velocity in meters per second.

$L(j)$  = Total gun lead angle.

#### 4.2 Projectile Velocity:

$V(a)$  = Average projectile velocity, varies with the selected ammunition (one of five).

Magnitude: \_\_\_\_\_ m/sec (800m/sec used for computer sizing)

Accuracy: 1%

#### 4.3 Combined Measurement and Dispersion Variances:

$\sigma(TW)^2$  = Combination of the horizontal or transverse round-to-round dispersion variance and the measurement variance. Measured transverse to the line-of-sight to target.

Magnitude:  $(3 \text{ mils})^2$

Accuracy: 0.1%

$\sigma(EW)^2$  = Combination of the elevation round-to-round dispersion variance and the measurement variance. Measured in the elevation plane.

Magnitude:  $(3 \text{ mils})^2$

Accuracy: 0.1%

$\sigma(VW)^2$  = Combined equivalent angle variance for the muzzle velocity, round-to-round dispersion and measurement error.

Magnitude:  $(4 \text{ mils})^2$

Accuracy: 0.1%

#### 4.4 Data Storage of Matrices: All temporary storage

$U(j)$ :  $n$  values must be stored, corresponding to the total number of observations that can be made and processed during the predicted time of flight  $t(pj)$  for the just observed projectile. The maximum value for  $n$  is 100 based upon a measurement every interrupt time (every 0.1 sec) and the maximum allowable flight time of 10 seconds.

K(j): Last value only

M(j) = P(j-1): Last value only

X(j): Last value only

## 5. Operations Performed

### 5.1 Defined Matrices

a) Measurement Matrix

$$Z(j) = \begin{pmatrix} \Delta T(j) \\ \Delta E(j) \end{pmatrix}$$

where  $\Delta T(j)$  and  $\Delta E(j)$  are the input parameters for the residual miss distance measurements (Item 3.1)

b) A Priori Bias Error Covariance Matrix

$$M(1) = P(0) = \begin{pmatrix} \sigma^2(A) & 0 & 0 \\ 0 & \sigma^2(E) & 0 \\ 0 & 0 & \sigma^2(V) \end{pmatrix}$$

where the non-zero terms are the bias variances of Item 4.1.

c) Random Error Matrix

$$\begin{aligned} R(j) &= \langle V(j) \ V(j)^T \rangle \\ &= \langle G(j) \ W(j-tp) \ G^T(j) \ W(j-tp)^T \rangle \\ &= \begin{pmatrix} \sigma^2(TW) + \sigma^2(VW) \cos^2 \underline{E}(mj) \sin^2 \delta(pj) & \sigma^2(VW) \sin \sigma(pj) \cos \underline{E}(mj) \sin \delta(pj) \\ \sigma^2(VW) \sin \sigma(pj) \cos \underline{E}(mj) \sin \delta(pj) & \sigma^2(EW) + \sigma^2(VW) \sin^2 \sigma(pj) \end{pmatrix} \end{aligned}$$

where the scalar parameters are defined in Items 3.2, 3.3, and 4.2.

d) Bias Geometry Matrix

$$H(j) = \begin{pmatrix} \cos \underline{E}(mj) & 0 & \cos \underline{E}(mj) \sin \delta(pj) \\ 0 & 1 & \sin \sigma(pj) \end{pmatrix}$$

where the scalar terms are input parameters defined in Items 3.2 and 3.3.

e) Bias Error State Vector Matrix

The terms of this matrix constitute the output of this block.

$$X(j) = \begin{pmatrix} A(bj) \\ E(bj) \\ v(bj) \end{pmatrix}$$

where:  $A(bj)$  = Azimuth angle bias correction factor at time  $t(j)$

$E(bj)$  = Elevation angle bias correction factor at time  $t(j)$

$v(bj)$  = Angular equivalent of the muzzle velocity bias correction factor at time  $t(j)$

The angular equivalent of the muzzle velocity bias is defined as the angular miss along the projectile's trace across the sensor's field of view. Its relationship to the actual muzzle velocity bias is given by the note in Item 4.1 above.

5.2 Algebraic Operations:

5.2.1 Predicted Projectile Time of Flight:

- a) Retrieve from the Ballistic Data Store (Block XIII, Item 4.1) the predicted projectile times of flight,  $t(p,n)$  and  $t(p,n + 1)$ , for projectiles fired at clock times  $t(n)$  and  $t(n + 1)$ . This is done using the table entry indices of input Item 3.5.
- b) Compute the predicted projectile time of flight for the equal range value by

$$t(pj) = C(j) [t(p,n + 1) - t(p,n)] + t(p,n)$$

5.2.2 Number of Miss Measurements:

- a) Compute the number of sampling periods made during the previous  $t(pj)$  seconds where  $t(pj)$  is the predicted projectile time of flight for the last observed projectile (Item 5.2.1, Step b). Since the sampling rate in this AFAADS analysis is 10 times per second.

$$p = 10 t(pj)$$

It should be noted that  $p$  is also the maximum number of miss measurements that can be made.

- b) Actual number of miss measurements,  $n$ , is the number of sampling intervals during the last  $p$  intervals when projectile miss data was received and processed.

### 5.3 Matrix Operations:

These operations are performed each time a projectile miss measurement is received. For time  $t(j)$ :

- a) Compute matrices  $Z(j)$ ,  $R(j)$ , and  $H(j)$ .
- b) Compute the Kalman Gain.

$$K(j) = M(j) H(j)^T [H(j) M(j) H(j)^T + R(j)]^{-1}$$

- c) Update the Covariance Matrix (2 forms exist)

$$M(j+1) = P(j) = \begin{cases} M(j) - K(j) [H(j) M(j) H(j)^T + R(j)] K(j)^T \\ \text{or} \\ M(j) [I - H(j)^T K(j)^T] \end{cases}$$

- d) Update the Bias Correction Matrix

$$U(j) = -K(j) [Z(j) + H(j) \sum_{k=1}^n U(j-k)]$$

where:  $n$  = number of miss measurements made in flight time of projectile,  $t(pj)$ . (See Item 5 2.2, Step b).

- e) Compute the Bias Error State Vector Matrix

$$X(j) = X(j-1) + U(j)$$

The terms of this matrix constitute the output parameters.

- f) Store the latest values of  $U(j)$ ,  $K(j)$ ,  $M(j)$ , and  $X(j)$  (See Item 4.4).

6. Outputs: The algebraic sign of these correction factors will remove the indicated biases.

6.1 Muzzle Velocity Bias Correction Factor: To Block XXII

V(bj) = Muzzle velocity bias correction factor expressed as an angular miss along the projectile's trace across the sensor's field of view.

Magnitude: -25 mils to +25 mils

Accuracy: 0.1%

6.2 Angle Bias Correction Factors: To Block XXXVII

A(bj) = Azimuth angle bias correction factor, applied to gun azimuth lead angle; positive clockwise.

Magnitude: -2 to +2 deg.

Accuracy: 0.1%

E(bj) = Elevation angle bias correction factor applied to gun elevation angle; positive up.

Magnitude: -2 to + 2 deg.

Accuracy: 0.1%

7.0 Comments:

7.1 Matrix Inversion: Only a 2 x 2 matrix is involved.

$$\begin{pmatrix} a(11) & a(12) \\ a(21) & a(22) \end{pmatrix}^{-1} = \frac{\begin{pmatrix} a(22) & -a(12) \\ -a(21) & a(11) \end{pmatrix}}{[a(11)a(22) - a(12)a(21)]}$$

7.2 Sample Computation:

Appendix E contains a sample calculation of the algorithms in this block.

8. References:

a) Volume I of this report, Section 3.

b) Appendix E for a sample calculation.

Block XVII: Stop Miss Distance Computation

1. Function: Processing branch point where projectile miss distance computations are not processed until the next sampling cycle.

2. Frequency: Once upon receipt of a stop miss distance computation trigger, no more frequent than every 0.1 second.

3. Inputs:

3.1 No Miss Distance Computation Trigger: From Blocks X and XXIV, trigger signal that no valid miss distance data is present for processing.

4. Constants: None

5. Operations Performed: Computer processing stops all miss distance measurement computations until next sample interval.

6. Outputs: Signal to Executive relative to applicable applications programs.

7. Comments: None

Block XVIII: Wind Data (from Operator)

1. Function: To provide wind data for ballistic correction, as inputted by the Operator during Initialization I Mode.

2. Frequency:

2.1 Operator Panel Polling: Operator's Panel is polled for a change in wind data every 0.1 second during Initialization I Mode.

2.2 Ballistic Corrections: Wind data to be available for ballistic corrections during the Initialization I Mode and for every ballistic computation update, i. e., every 0.1 second, during the Real Time Fire Control Mode.

3. Inputs:

3.1 Polling Trigger: From Block XXIX, Real Time Clock, the polling trigger is received every 0.1 second for polling the output buffer of the Operator's Panel (peripheral device D1).

3.2 Operator's Panel: From peripheral device D1, the value of the wind is read This can be either:

- a. PRESET value trigger
- b. ENTERED wind speed and heading

W = Wind speed

Magnitude: 0m/sec to 30m/sec (66 mph)

Accuracy: 1.0m/sec.

$\theta(r)$  = Relative wind direction towards AFAADS, measured clockwise from front of vehicle

Magnitude: 0 deg to 360 deg.

Accuracy: 5 deg.

4. Constants:

4.1 Preset Value: Value to be used if Operator enters PRESET

Speed: Magnitude: 0m/sec

Accuracy: 1.0m/sec

Relative Direction: Magnitude: 0 deg (actually undetermined)

Accuracy: 5 deg.

4.2 Stored Value: Last value entered by the Operator

W = Magnitude of entered wind speed

Magnitude: 0m/sec to 30m/sec

Accuracy: 1.0m/sec

$\theta(r)$  = Relative direction of the wind toward AFAADS, measured clockwise from front of vehicle.

Magnitude: 0 deg to 360 deg.

Accuracy: 5 deg.

5. Operations Performed:

Upon receipt of the Polling Trigger from Block XXIX, the output buffers for wind on the Operator's Panel (part of peripheral device D1) are polled for any change.

- a. If no change, processing ends.
- b. If changed and operator has entered PRESET, enter Preset Value of wind (Item 4.1) in Stored Value memory location (Item 4.2).
- c. If changed and operator has entered a wind velocity and/or direction, enter this or these new values in the Stored Value memory locations. Note both direction and speed must be either Preset or User values. A mix is invalid.

The Stored Value memory wind values (Item 4.2) shall always be ready for use in ballistic corrections during both Initialization I and Real Time Fire Control Modes.

6. Outputs:

6.1 Wind: To Block IV during the Initialization Mode and to Block V during the Semi-Automatic Real Time Fire Control Mode:

W = Magnitude of stored wind

$\theta(r)$  = Relative direction of the stored wind.

The magnitudes and accuracies of these quantities are the same as Section 4 2, Stored Values.

7. Comments: None

Block XIX: Air Data (from Operator)

1. Function: To provide air temperature and air pressure for ballistic corrections, as inputted by the Operator during Initialization I Mode.

2. Frequency: Operator's Panel is polled for a change in air temperature and/or air pressure every 0.1 second during Initialization I Mode.

3. Inputs:

3.1 Polling Trigger: From Block XXIX, Real Time Clock, the polling trigger is received every 0.1 second for polling the output buffer of the Operator's Panel (peripheral device D1).

3.2 Operator's Panel: From the output buffer of the Operator's Panel, part of peripheral device D1, the values of air temperature and air pressure are read. Values are:

a. PRESET value trigger for either air temperature or air pressure.

b. Air Temperature Value, T: Value entered by operator.

Magnitude: -20 deg to +120 deg F.

Accuracy: 1 deg F

c. Air Pressure Value, P: Value entered by operator

Magnitude: 500 mb to 1040 mb

Accuracy: 1 mb

4. Constants:

4.1 Preset Value: Value to be used if Operator enters PRESET

Air Temperature: Magnitude: 70 deg F.

Air Pressure: Magnitude: 1013 mb.

4.2 Stored Values: Last value entered by the Operator:

Air Temperature: Magnitude: -20 deg to +120 deg F

Accuracy: 1 deg F

Air Pressure: Magnitude: 500 mb to 1040 mb

Accuracy: 1 mb

5. Operations Performed:

Upon receipt of the Polling Trigger from Block XXIX during Initialization I Mode, the output buffers for air temperature and air pressure on the Operator's Panel are polled for change:

- a. If no change, processing end.
- b. If changed and PRESET has been entered, enter appropriate Preset Value (Item 4.1) in Stored Value memory location (Item 4.2).
- c. If changed and a value has been entered, enter that value in the appropriate Stored Value memory location.

For cases b. and c., the new values are sent to Block IV Ballistic Computations, for correcting the coefficients of the ballistic equations during Initialization I Mode.

6. Outputs: To Block IV during Initialization I Mode only. All new values of air temperature and/or air pressure. Magnitudes and accuracies as given in 4.2 above.

7. Comments: None

Block XX: Coordinate Conversion to Polar (Position)

1. Function: To convert current smoothed target position data from Cartesian coordinates,  $X(j)$ ,  $Y(j)$ ,  $Z(j)$ , to polar coordinates,  $D(mj)$ ,  $A(mj)$ ,  $E(mj)$ .

2. Frequency: Every update of the current target position, i. e., every 0.1 second.

3. Inputs:

3.1 Current Smoothed Target Position: From Block II, current smoothed target position in Cartesian coordinates for flat earth.

$\underline{X}(j)$  = Cross position of target relative to AFAADS vehicle at time  $t(j)$ , right side is positive.

Magnitude: -10,000m to +10,000m

Accuracy: \_\_\_\_\_ m

$\underline{Y}(j)$  = Lengthwise position of target relative to AFAADS vehicle at time  $t(j)$ , positive forward.

Magnitude: -10,000m to +10,000m

Accuracy: \_\_\_\_\_ m

$\underline{Z}(j)$  = Altitude of target above AFAADS vehicle at time  $t(j)$ , positive up.

Magnitude: -900m to +8,000m

Accuracy: \_\_\_\_\_ m

4. Constants: None

5. Operations Performed:

For every update of current target position, compute

$$\underline{D}(mj) = [\underline{X}^2(j) + \underline{Y}^2(j) + \underline{Z}^2(j)]^{1/2}$$

$$\underline{E}(mj) = \sin^{-1}[\underline{Z}(j)/\underline{D}(mj)]$$

$$\underline{A}(mj) = \begin{cases} \sin^{-1}[\underline{X}(j)/\underline{D}(mj) \cos \underline{E}(mj)] \\ \text{or} \\ \cos^{-1}[\underline{Y}(j)/\underline{D}(mj) \cos \underline{E}(mj)] \end{cases}$$

6. Outputs:

6.1 Range: To Blocks XII, XXIV, and XXVI

$\underline{D}(mj)$  = Slant range from AFAADS gun to current target position at time  $t(j)$ , positive out.

Magnitude: 100m to 10,000m

Accuracy: \_\_\_\_\_ m.

6.2 Azimuth: To Blocks XXVI, XXVIII, XXXVI, and XXXVII.

$\underline{A}(mj)$  = Current target azimuth from front of vehicle at time  $t(j)$ , measured clockwise.

Magnitude: 0 deg to 360 deg.

Accuracy: \_\_\_\_\_ mils.

6.3 <u>Elevation:</u>	To Blocks:	XVI	XXVIII
		XXII	XXXVI
		XXIII	XXXVII
		XXVI	

$E(mj)$  = Current target elevation at time  $t(j)$ , positive up.

Magnitude: -5 deg to +85 deg.

Accuracy: \_\_\_\_\_ mils.

7. Comments: Any other equivalent set of coordinate conversion equations can be used.

#### Block XXI: Lead Angle Extraction and Computation

##### 1. Function:

- a. To extract from Ballistic Data Store (Block XIII) the predicted gun lead angles for time  $t(n)$  and  $t(n + 1)$ .
- b. To interpolate to the predicted equal range point, values of the gun lead angles.
- c. To compute the total lead angle.

2. Frequency: Whenever a valid projectile miss distance measurement is made, maximum frequency of 10 times per second.

##### 3. Inputs:

##### 3.1 Matched Time Data: From Block XIV

1. Indices of table entries in Block XIII which provide best time matches above and below time of projectile fire,  $t(j-p)$ , namely at times  $t(n)$  and  $t(n + 1)$ .
2.  $C(j)$  = Corrective factor for equal range point correction.  
Magnitude: 0 to 1.0, a numeric  
Accuracy: 0.05

##### 3.2 Target Lead Angles: From Block XIII

$\delta(p,n), \delta(p,n + 1)$  = Predicted gun lead in azimuth for projectile fired at times  $t(n)$  and  $t(n + 1)$ ; positive clockwise.  
Magnitude: -90 deg to +90 deg.  
Accuracy: 0.25 mil

$\sigma(p,n), \sigma(p,n + 1) =$  Predicted gun elevation lead angles for projectiles fired at times  $t(n)$  and  $t(n + 1)$ ; positive up.

Magnitude: -30 to +70 deg.

Accuracy: 0.25 mil.

4. Constants: None

5. Operations Performed:

5.1 Extract:

With the table indices corresponding to table times  $t(n)$  and  $t(n + 1)$ , extract from Block XIII, the Ballistic Data Store, the corresponding values of  $\delta(p,n), \sigma(p,n), \delta(p,n + 1), \sigma(p,n + 1)$ .

5.2 Interpolate:

Determine by linear interpolation the equal range or predicted values for the lead angles

$$\delta(pj) = C(j) [\delta(p,n + 1) - \delta(p,n)] + \delta(p,n)$$

$$\sigma(pj) = C(j) [\sigma(p,n + 1) - \sigma(p,n)] + \sigma(p,n)$$

5.3 Total Lead Angle:

Compute total lead angle:

$$L(j) = \sqrt{\delta^2(pj) + \sigma^2(pj)}$$

6. Outputs:

6.1 Request: To Block XIII

Table indices corresponding to times  $t(n)$  and  $t(n + 1)$  with request for gun lead angle data,  $\delta(p,n), \sigma(p,n), \delta(p,n + 1)$  and  $\sigma(p,n + 1)$ .

6.2 Predicted Gun Lead Angles: To Block XVI

$L(j) =$  Total gun lead for projectile arriving at target at time  $t(j)$ .

Magnitude: 0 deg to +90 deg.

Accuracy: 0.25 mil.

$\delta(pj) =$  Azimuth gun lead angle for projectile arriving at target at time  $t(j)$ , positive clockwise.

Magnitude: -90 deg. to +90 deg.

Accuracy: 0.25 mil.

$\sigma(pj)$  = Elevation lead angle for projectile arriving at target at time  $t(j)$ , positive up.

Magnitude: -30 deg. to +70 deg.

Accuracy: 0.25 mil.

7. Comments: None

#### Block XXII: Muzzle Velocity Bias Correction

1. Function: To apply the muzzle velocity bias correction factor to the gun's azimuth and elevation lead angles. This correction factors is a function of the type of ammunition and is updated by the projectile miss distance measurements.

2. Frequency: Every update of the gun train and elevation orders, i.e., every 0.1 second.

3. Inputs:

3.1 Ammunition Selection: From Block XXXIV, the selected ammunition type, 5 types.

3.2 Ballistically Corrected Gun Lead Angles: From Block XXXVI:

$\delta(bcj)$  = Ballistically corrected gun lead angle in azimuth; positive clockwise from target's azimuth.

Magnitude: -90 deg. to +90 deg.

Accuracy: 0.25 mil.

$\sigma(bcj)$  = Ballistically corrected gun lead angle in elevation; positive above target.

Magnitude: -30 deg. to +70 deg.

Accuracy: 0.25 mil.

3.3 Muzzle Velocity Bias Correction Factor: From Block XVI:

$V(bj)$  = Muzzle velocity bias correction factor expressed as an angular miss along the projectile's trace across the sensor's field of view.

Magnitude: -25 mils to +25 mils.

Accuracy: 0.1%.

3.4 Target Elevation: From Block XX

$\underline{E}(mj)$  = Current smoothed target elevation angle measured from the AFAADS vehicle; positive up.  
Magnitude: -5 deg. to +85 deg.  
Accuracy: \_\_\_\_\_ mils.

3.5 Target's Angular Velocity: From Block XXVI

$\dot{\underline{I}}(j)$  = Current smoothed target transverse angular velocity relative to the line of sight; positive clockwise.  
Magnitude: -90 deg/sec to +90 deg/sec.  
Accuracy: \_\_\_\_\_ mils/sec.

$\dot{\underline{E}}(j)$  = Current smoothed target elevation angular velocity relative to the line of sight; positive up.  
Magnitude: -25 deg/sec to +25 deg/sec.  
Accuracy: \_\_\_\_\_ mils/sec.

4. Constants:

4.1 Permanent Memory:

$g$  = Acceleration due to gravity  
Magnitude: 9.80665 m/sec<sup>2</sup>.  
Accuracy: To be compatible with other parameters.

$V(a)$  = Average projectile velocity. Varies with the selected ammunition (5 types). Constant is also used in Block XVI.  
Magnitude: \_\_\_\_\_ m/sec (800m/sec used for computer sizing task)  
Accuracy: 1%.

4.2 Temporary Memory: Store the last used values of the gun lead angle corrections in azimuth and elevation by ammunition type. (For each of five projectile types, store one azimuth and one elevation angle for a total of 10 values)

5. Operations Performed:

The logic flow is shown on Figure 11-9. Each of the following steps are correlated to blocks on the figure.

a) Correction Logic Determination: On every update of the gun train

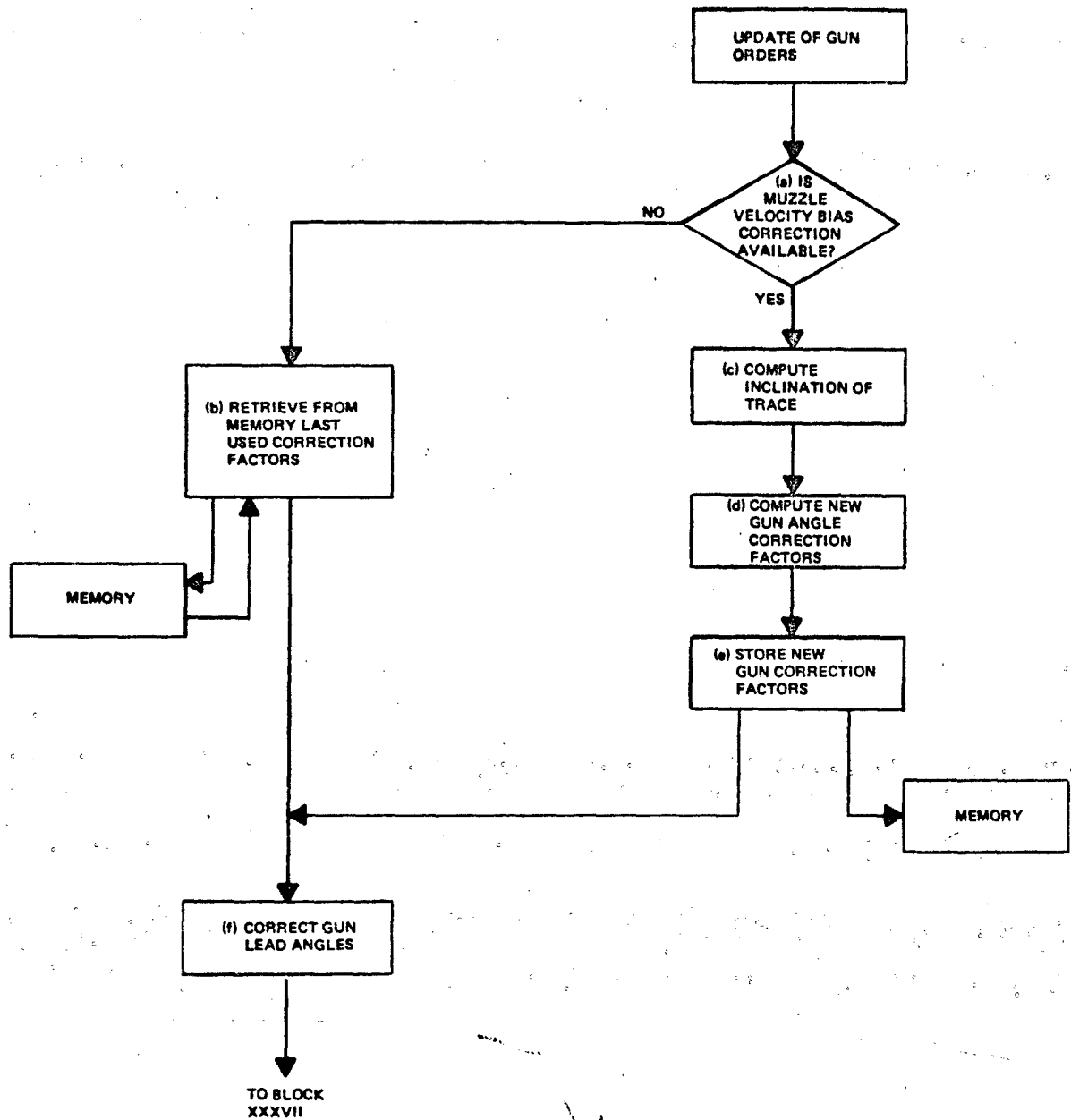


Figure 11-9. Bias Muzzle Velocity Correction Logic.

and elevation orders, determine if a muzzle velocity bias correction factor is available from Block XVI. If no factor available, proceed to Step b; if available, proceed to Step c.

- b) Data Retrieval: If no new correction factor is available, retrieve from memory (Item 4.2) the last used gun lead angle correction factors in azimuth and elevation for the ammunition type being used. Then proceed to Step f.
- c) Observed Inclination of Projectile's Trajectory: If a new muzzle velocity correction factor is available (Item 3.3), compute the observed inclination of the trace of the projectile's trajectory across the field-of-view of the sensor for the selected ammunition type.

$$\tan \psi(j) = \{ \dot{E}(j) - [g/V(a)] \sin E(mj) \} / \dot{T}(j)$$

Proceed to Step d.

- d) New Correction Angles: Compute new gun lead angle correction factors in azimuth and elevation due to muzzle velocity bias by:

$$\text{Azimuth: } \delta(vj) = V(bj) \cos \psi(j)$$

$$\text{Elevation: } \sigma(vj) = V(bj) \sin \psi(j)$$

Proceed to Step e.

- e) Store: For the selected ammunition type, replace the gun lead angle correction factors in azimuth and elevation by those computed in Step d. Proceed to Step f.
- f) Lead Angle Corrections: Apply the gun lead angle correction factors (from Step b or Step d) to the ballistically corrected lead angles (Item 3.2).

$$\text{Azimuth: } \delta(vcj) = \delta(bcj) + \delta(vj)$$

$$\text{Elevation: } \sigma(vcj) = \sigma(bcj) + \sigma(vj)$$

Output these gun lead angles.

6. Outputs:

6.1 Velocity Corrected Gun Lead Angles: To Block XXXVII

$\delta(vcj)$  = Muzzle velocity corrected gun lead angle (gun leads target) in azimuth; positive clockwise.

Magnitude: -90 deg to +90 deg.

Accuracy: \_\_\_\_\_ mils.

$\sigma(vcj)$  = Muzzle velocity corrected gun lead angle in elevation; positive up.

Magnitude: -30 deg to +70 deg.

Accuracy: \_\_\_\_\_ mils.

7. Comments: None

Block XXIII: Coordinate Conversion to LOS Transverse

1. Function: To change the azimuth target maneuver correction to the line-of-sight (LOS) transverse target maneuver correction.

2. Frequency: Whenever a valid projectile miss distance measurement is made, i. e., no more frequent than every 0.1 second.

3. Inputs:

3.1 Target Elevation: From Block XX.

$\underline{E}(mj)$  = Current smoothed target elevation at time  $t(j)$ , positive up.

Magnitude: -5 to +85 deg.

Accuracy: \_\_\_\_\_ mils.

3.2 Azimuth Miss due to Target Maneuver: From Block XXVIII

$\Delta A(vj)$  = Azimuth miss angle due to target maneuver at time  $t(j)$ , positive for a target to the right of the predicted position.

Magnitude: -45 deg to +45 deg.

Accuracy: \_\_\_\_\_ mils.

4. Constants: None

5. Operations Performed: Coordinate change.

$\Delta T(vj) = \Delta A(vj) \cos \underline{E}(mj)$

6. Output:

6.1 Transverse Target Maneuver Correction: To Block XXXVIII

$\Delta T(vj)$  = Target maneuver correction in a plane through the line-of-sight to the target and perpendicular to the elevation plane, i. e., the transverse target maneuver correction angle; positive for target to right of predicted position.  
Magnitude: -45 deg to +45 deg.  
Accuracy: \_\_\_\_\_ mils.

7. Comments: None

Block XXIV: Is Miss Distance Data Valid?

1. Function: To determine whether the measured angular miss distance data will provide valid data for making bias error corrections.

2. Frequency: Every time miss distance data is received, at most every 0.1 second.

3. Inputs:

3.1 Measured Miss Distance Data: From Block X.

$\Delta T(mk)$  = Measured transverse projectile miss angle at time  $t(k)$ , positive if the miss is to the right.

Magnitude: -4 deg to +4 deg except -0.008 deg. to +0.008 deg can not be resolved.

Accuracy: 0.25 mil.

$\Delta E(mk)$  = Measured elevation projectile miss angle at time  $t(k)$ , positive if the miss is up.

Magnitude: -2 deg to +2 deg except -0.008 deg. to +0.008 can not be resolved.

Accuracy: 0.25 mil.

$\Delta D(mk)$  = Measured projectile-to-target range at time  $t(k)$ , positive for projectile between AFAADS and target.

Magnitude: -600m to +600m.

Accuracy: 2m.

$t(k)$  = Clock time when measurement was made.

Magnitude: 0 to about 5 min.

Accuracy: 0.005 sec.

3.2 Target Angular Rate: From Block XXV

$\underline{\omega}(k)$  = Magnitude of total angular velocity of target relative to AFAADS gun at time  $t(k)$ .

Magnitude: 0 to 95 deg/sec.

Accuracy: \_\_\_\_\_ mils/sec.

3.3 Target Range Rate: From Block XXXVI

$\dot{D}(k)$  = Current smoothed target range rate relative to AFAADS gun at time  $t(k)$ ; opening rate positive.

Magnitude: -400m/sec to +400m/sec.

Accuracy: \_\_\_\_\_ m/sec.

3.4 Target Range: From Block XX

$\underline{D}(mj)$  = Range to current target position at time  $t(j)$ .

Magnitude: 100m to 10,000m.

Accuracy: \_\_\_\_\_ m.

3.5 Projectile Type: From Block XXXIV, status word indicating the selected projectile type.

4. Constants:

4.1 Permanent Storage: The following parameters are stored in the Program Memory (PROM or ROM).

$\dot{D}(p)$  = Table of projectile velocities as a function of target range and projectile type.

Range of Magnitudes: Minimum: 300m/sec

Maximum: 1570m/sec.

$\Delta t$  = Data sampling period, determined by Block XXIX

Magnitude: 0.1 second.

$D(d)$  = Range dispersion due to muzzle velocity dispersion times time-of-flight. (Depends on ammunition type, 5 considered).

Magnitude: \_\_\_\_\_ m (60m used in computer sizing task, see Appendix D)

$\omega(o)$  = Minimum angular velocity constant; equal to angular miss distance error divided by half the scan period.

Magnitude: 16 mils/sec = 0.92 deg/sec.

#### 4.2 Temporary Storage:

The last two time samples of each of the seven input quantities.

#### 5. Operations Performed:

The logic operations presented are based upon a high pulse rate laser operating with the FLIR. As discussed in Volume I, Section 3.2, such a combination provides range gating of the FLIR returns; and thus the combination is similar to radar. If such a combination is not feasible in the AFAADS time period, either the closed loop bias corrections in AFAADS could be changed to the two dimensional case with a reduction in the processing requirements or a radar tracking sensor could be considered.

Proceeding on the above basis, Figure 11-10 shows the logic flow. It is amplified by the following:

a) Read in:

- 1) From Block X:  $\Delta T(mk)$ ,  $\Delta E(mk)$ ,  $\Delta D(mk)$ , and  $t(k)$ .
- 2) From Block XXV:  $\underline{w}(k)$
- 3) From Block XXVI:  $\underline{D}(k)$

Proceed to Step b).

b) Store data in memory, replacing the older of the two sets of data. Go to Step c).

c) Compare the clock times  $t(k)$  and  $t(q)$  of the two sets of data in memory to determine if their difference equals the data sampling period,  $\Delta t$ : i. e.,

$$\text{Does } |t(k) - t(q)| = \Delta t?$$

If yes, go to Step d). If no, go to Step e).

d) Since miss data was detected on consecutive time samples,

- i) Select the lesser of  $\Delta D(mk)$  and  $\Delta D(mq)$ .
- ii) In the following use that file in the temporary storage, Item 4.2, that contains the lesser of  $\Delta D(mk)$  and  $\Delta D(mq)$ . Go to Step f).

e) Since miss data was detected on only the last time sample, use that file in the following. Proceed with Step f).

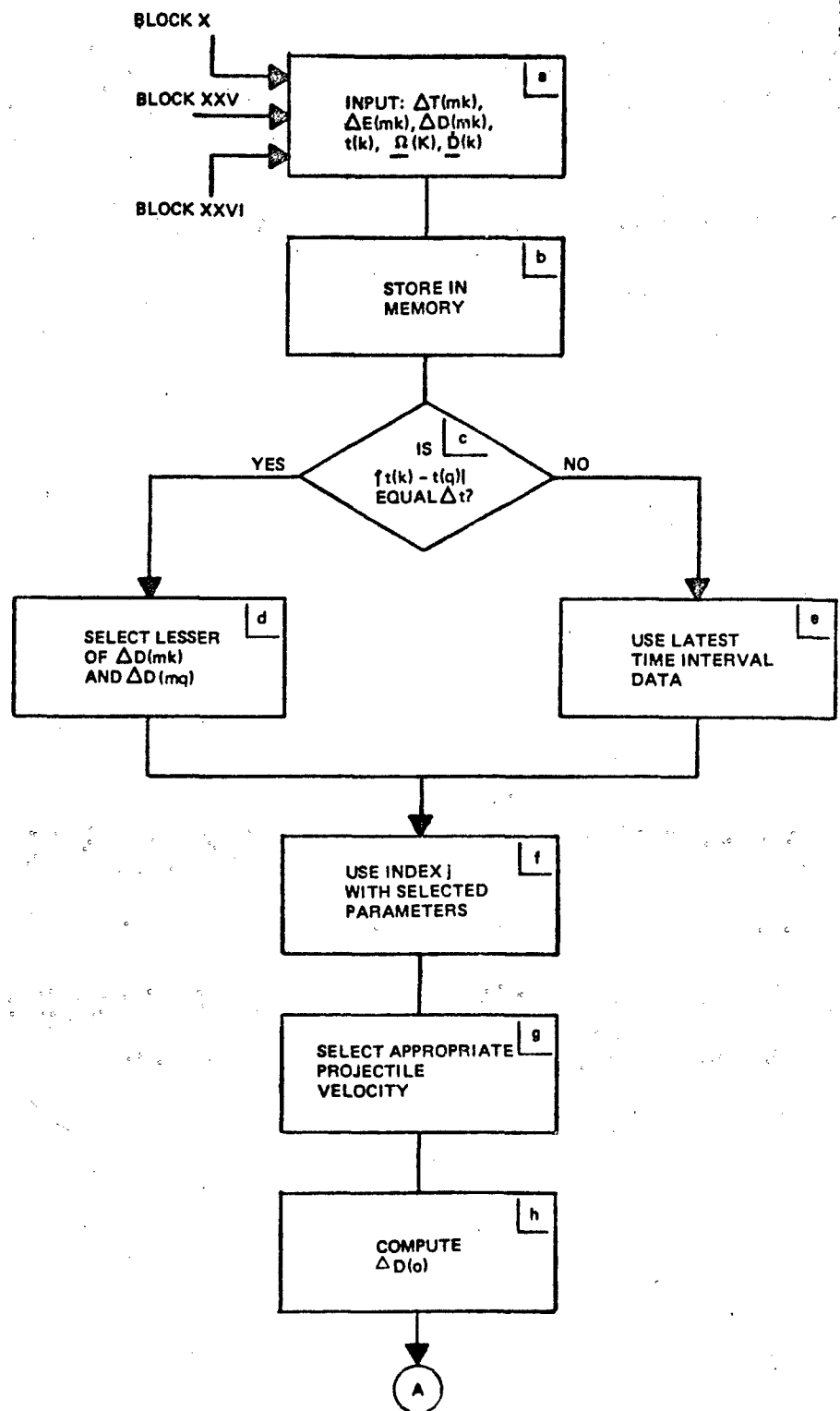


Figure 11-10. Logic Flow for Miss Distance Data Determination (Sheet 1 of 2).

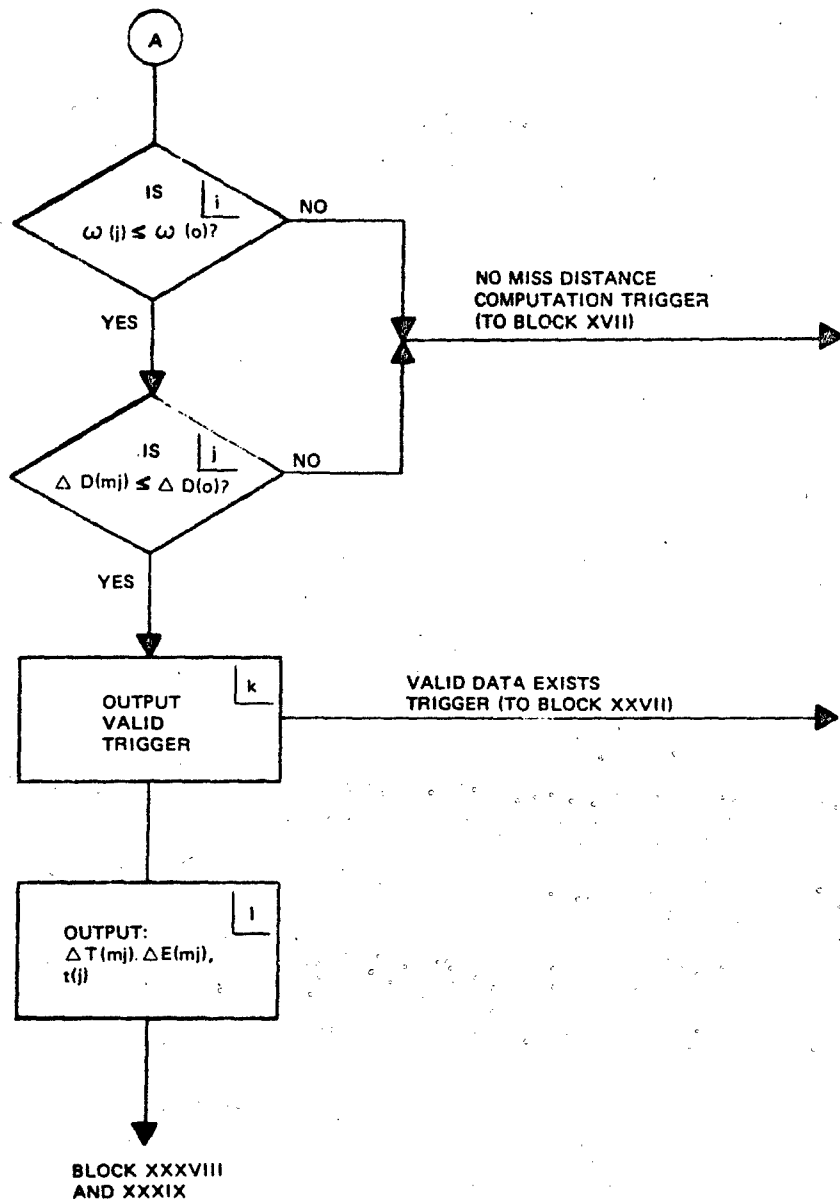


Figure 11-10. Logic Flow for Miss Distance Data Determination (Sheet 2 of 2).

- f) For convenience and clarity, the selected data will have the time interval designated by j rather than either k or q. Go to Step g.
- g) Based upon the projectile type (Item 3.5) and the current target range (Item 3.4), select from memory the appropriate projectile velocity (Item 4.1). Go to Step h).
- h) Compute
 
$$\Delta D(o) = [\dot{D}(p) - \dot{D}(j)] \Delta t + D(d)$$
 Go to Step i).
- i) Is  $\omega(j) \leq w(o)$ ?
  - If no, stop computations and output No Miss Distance Computation Trigger to Block XVII.
  - If yes, continue with Step j).
- j) Is  $\Delta D(mj) \leq \Delta D(o)$ ?
  - If no, stop computations and output No Miss Distance Computation Trigger to Block XVII.
  - If yes, valid projectile miss data has been received and the remaining computations are to proceed, Step k).
- k) Generate Valid Data Exists Trigger for Block XXVII.
- l) Output to Block XXXVIII:  $\Delta T(mj), \Delta E(mj), t(j)$

## 6. Outputs:

6.1 Valid Data Exists Trigger: To Block XXVII, send a trigger that valid projectile miss data has been received and hence the bias correction algorithm should proceed.

6.2 Projectile Miss Data: To Block XXXVIII:

$\Delta T(mj)$  = Measured transverse projectile miss angle.  
(Identical to input)

$\Delta E(mj)$  = Measured elevation projectile miss angle.  
(Identical to input)

6.3 Time of Projectile Miss Measurement: To Block XXXIX

$t(j)$  = Time of day when measurement was made.  
(Identical to input)

6.4 No Miss Distance Computation Trigger: To Block XVII, send No Miss Distance Computation Trigger to stop all miss distance computations for this sampling period.

7. Comments: None

8. References:

- a) Volume I, Section 3.
- b) Appendix D.

Block XXV: Total LOS Angular Velocity

- 1. Function: To compute the total angular velocity of the line-of-sight from the AFAADS gun to the target.
- 2. Frequency: Every update of the current target position, i. e., every 0.1 second.

3. Inputs:

3.1 Current Target Angular Rates: From Block XXVI.

$\dot{I}(j)$  = Current smoothed target transverse angular rate relative to AFAADS gun at time  $t(j)$ , positive to the right.  
Magnitude: -90 deg/sec to +90 deg/sec.  
Accuracy: \_\_\_\_\_ mils/sec.

$\dot{E}(j)$  = Current smoothed target elevation angular rate relative to AFAADS gun at time  $t(j)$ , positive up.  
Magnitude: -25 deg/sec to +25 deg/sec.  
Accuracy: \_\_\_\_\_ mils/sec.

4. Constants: None

5. Operations Performed: Upon receipt of every update of the current target position, compute the magnitude of the total angular velocity:

$$\omega(j) = \sqrt{\dot{I}(j)^2 + \dot{E}(j)^2}$$

6. Output:

6.1 Total Angular Velocity: To Block XXIV.

$\omega(j)$  = Magnitude of the total angular velocity of the target relative to the AFAADS gun at time  $t(j)$   
Magnitude: 0 to 95 deg/sec.  
Accuracy: \_\_\_\_\_ mils/sec.

7. Comments: None

Block XXVI: Coordinate Conversion to Polar (Velocity)

1. Function: To convert the smoothed target velocity data from Cartesian coordinates into polar coordinates.

2. Frequency: Every update of the current target position, i. e., every 0.1 second.

3. Inputs:

3.1 Current Smoothed Target Velocity: From Block II

$\dot{X}(j)$  = Current smoothed cross velocity of target relative to AFAADS vehicle at time  $t(j)$ , right side is positive.

Magnitude: -400m/sec to +400m/sec.

Accuracy: \_\_\_\_\_ m/sec.

$\dot{Y}(j)$  = Current smoothed length se velocity of target relative to AFAADS vehicle at time  $t(j)$ , positive forward.

Magnitude: -400m/sec to +400m/sec.

Accuracy: \_\_\_\_\_ m/sec.

$\dot{Z}(j)$  = Current smoothed climb rate of target relative to AFAADS vehicle at time  $t(j)$ , positive up.

Magnitude: -400m/sec to +400m/sec.

Accuracy: \_\_\_\_\_ m/sec.

3.2 Current Smoothed Target Position: From Block XX

$D(mj)$  = Smoothed slant range from AFAADS gun to current target position at time  $t(j)$ .

Magnitude: 100m to 10,000m

Accuracy: \_\_\_\_\_ m/sec.

$A(mj)$  = Smoothed target azimuth from AFAADS gun measured from front of vehicle in a clockwise direction.

Magnitude: 0 to 360 deg.

Accuracy: \_\_\_\_\_ mil.

$E(mj)$  = Smoothed target elevation from AFAADS gun; positive up.

Magnitude: -5 deg to +85 deg.

Accuracy: \_\_\_\_\_ mil.

4. Constants: None

5. Operations Performed: For every target position update, compute

$$\dot{\underline{I}}(j) = [\dot{\underline{X}}(j) \cos \underline{A}(mj) - \dot{\underline{Y}}(j) \sin \underline{A}(mj)] / \underline{D}(mj)$$

$$\dot{\underline{E}}(j) = \left\{ -[\dot{\underline{X}}(j) \sin \underline{A}(mj) + \dot{\underline{Y}}(j) \cos \underline{A}(mj)] \sin \underline{E}(mj) + \dot{\underline{Z}}(j) \sin \underline{E}(mj) \right\} / \underline{D}(mj)$$

$$\dot{\underline{D}}(j) = [\dot{\underline{X}}(j) \sin \underline{A}(mj) + \dot{\underline{Y}}(j) \cos \underline{A}(mj)] \cos \underline{E}(mj) + \dot{\underline{Z}}(j) \sin \underline{E}(mj)$$

6. Outputs:

- 6.1 Current Target Angular Rates: To Block XXV and XXII

$\dot{\underline{I}}(j)$  = Current smoothed target transverse angular rate relative to AFAADS gun at time  $t(j)$ , positive to the right.

Magnitude: -90 deg/sec to +90 deg/sec.

Accuracy: \_\_\_\_\_ mils/sec.

$\dot{\underline{E}}(j)$  = Current smoothed target elevation angular rate relative to AFAADS gun at time  $t(j)$ , positive up.

Magnitude: -25 deg/sec to +25 deg/sec.

Accuracy: \_\_\_\_\_ mils/sec.

- 6.2 Current Target Range Rate: To Block XXIV:

$\dot{\underline{D}}(j)$  = Current smoothed target range rate relative to AFAADS gun at time  $t(j)$ , opening rate positive.

Magnitude: -400m/sec to +400m/sec.

Accuracy: \_\_\_\_\_ m/sec.

7. Comments: None

8. Reference:

a) Reference 1, page 5-11.

Block XXVII: Initiate Projectile Miss Calculations

1. Function: To initiate the projectile miss calculations.

2. Frequency: Whenever a valid projectile miss distance measurement has been made, maximum frequency of 10 times per second.

3. Input:

3.1 Valid Data Exists Trigger: From Block XXIV, a trigger signal indicating that a valid projectile miss distance measurement has been made.

4. Constants: None

5. Operations Performed: Upon receipt of the trigger signal, the Executive will initiate the appropriate projectile miss measurement and bias correction algorithms.

6. Output:

6.1 Valid Miss Distance Trigger: To Block XII, a signal to execute the appropriate algorithm.

7. Comment: None

Block XXVIII: Target Maneuver Correction

1. Function: To determine the miss distances in azimuth and elevation due to target maneuvers.

2. Frequency: Whenever a valid projectile miss distance measurement is made, i. e., no more frequent than every 0.1 second.

3. Inputs:

3.1 Predicted Target Values: From Block XV

$A(p,j,t(j-p))$  = Predicted target azimuth angle at the equal range point, measured clockwise.

Magnitude: 0 to 360 deg.

Accuracy: \_\_\_\_\_ mils.

$E(p,j,t(j-p))$  = Predicted target elevation angle at the equal range point, positive up.

Magnitude: -5 to +85 deg.

Accuracy: \_\_\_\_\_ mils.

3.2 Current Target Values: From Block XX

$A(mj)$  = Current smoothed target azimuth at time  $t(j)$ , measured clockwise from front of vehicle.

Magnitude: 0 to 360 deg.

Accuracy: \_\_\_\_\_ mils.

$\underline{E}(mj)$  = Current smoothed target elevation at time  $t(j)$  above plane of vehicle, positive up.  
Magnitude: -5 deg. to +85 deg.  
Accuracy: \_\_\_\_\_ mils.

4. Constants: None

5. Operations Performed: Compute target maneuver correction.

$$\Delta A(vj) = \underline{A}(mj) - A(p, j, t(j-p))$$

$$\Delta E(vj) = \underline{E}(mj) - E(p, j, t(j-p))$$

6. Outputs:

6.1 Azimuth Miss Due to Target Maneuver: To Block XXIII

$\Delta A(vj)$  = Azimuth miss angle due to target maneuver at time  $t(j)$ , positive for target to right of predicted position.  
Magnitude: -45 deg to +45 deg.  
Accuracy: \_\_\_\_\_ mils.

6.2 Elevation Miss Due to Target Maneuver: To Block XXXVIII.

$\Delta E(vj)$  = Elevation miss angle due to target maneuver at time  $t(j)$ , positive for target above predicted position.  
Magnitude: -30 deg to +30 deg.  
Accuracy: \_\_\_\_\_ mils.

7. Comment: None

#### Block XXIX: Real Time Clock

1. Function: To provide timer interrupts for polling all input devices including the Operator's Panel. These interrupts are the major function controlling the software.

2. Frequency: For polling.

- a. Tracking Sensor (target data and rate gyro data): Every 0.1 sec.
- b. Tracking Sensor (projectile miss data): Every 0.1 sec.
- c. Operator's Panel: Every 0.1 sec.

3. Input:

3.1 Time of Day Request: From block XXX, request for time of day.

4. Constants: None

5. Operations Performed: Each of the following are logically independent.

5.1 Real Time Clock: Real time clock giving time of day

Scale Range: 0 to 5 minutes maximum

Accuracy: 0.005 sec.

5.2 Polling Triggers: Generate the required polling triggers at the required times for transmission.

5.3 Time-of-Day: Generate the time-of-day signal when requested.

6. Outputs:

6.1 Target Data Trigger: To Block VI, send trigger for polling the FLIR and laser sensors for target position data and the rate gyros for target angle rate data.

Frequency: Every 0.1 sec.

Accuracy: 0.005 sec.

6.2 Miss Data Trigger: To Block X, send trigger for polling the FLIR and laser sensors for projectile miss data.

Frequency: Every 0.1 sec.

Accuracy: 0.005 sec.

6.3 Poll Operator's Panel: To the Operator's Panel (Blocks IX, XVIII, XIX, XXXI, XXXIII and XXXIV), send trigger for polling any changes in the Operator's Panel (button pushes and/or keyboard entries).

Frequency: Every 0.1 sec.

Accuracy: 0.005 sec.

6.4 Time of Day Signal: To Block XXX

t(1) = Time of day at interval 1.

Magnitude: Clock time, 5 min. maximum

Accuracy: 0.005 sec.

7. Comment: None

#### Block XXX: Ballistic Data Input Register

1. Function: To assemble the data to be placed in the Ballistic Data Store, Block XIII.

2. Frequency: Every 0.1 second

3. Inputs:

3.1 Time Signal: From Block XXIX

$t(1)$  = Time of day at time interval 1.  
Magnitude: Clock time, 5 min. maximum  
Accuracy: 0.005 sec.

3.2 Projectile Time of Flight: From Block IV.

$t(p1)$  = Projectile time of flight for projectile fired at clock time interval 1.  
Magnitude: 0 sec. to 10 sec.  
Accuracy: 0.005 sec.

3.3 Predicted Target Position: From Block IV

$A(p1)$  = Predicted target azimuth,  $t(p1)$  seconds after clock time interval 1; measured clockwise.  
Magnitude: 0 deg. to 360 deg.  
Accuracy: \_\_\_\_\_ mils.

$E(p1)$  = Predicted target azimuth,  $t(p1)$  seconds after clock time interval 1; positive up.  
Magnitude: -5 deg. to +85 deg.  
Accuracy: \_\_\_\_\_ mils.

3.4 Gun Lead Angles: From Block XXXVII.

$\delta(1)$  = Corrected gun lead angle in azimuth for time interval 1; positive clockwise  
Magnitude: -90 deg. to +90 deg.  
Accuracy: 0.25 mil.

$\alpha(1)$  = Corrected gun lead angle in elevation for time interval 1; positive up.  
Magnitude: -30 deg. to +70 deg.  
Accuracy: 0.25 mil.

4. Constants: Buffer store for the six input quantities during assembly.

5. Operations Performed:

- a) Upon receipt of the projectile time of flight signal,  $t(p1)$ , from Block IV, interrogate the Real Time Clock (Block XXIX) for the time of day signal  $t(1)$ . Place both in the buffer store.
- b) Receive the predicted target position angles,  $A(p1)$  and  $E(p1)$ , and place in the buffer store.
- c) Receive the gun lead angles,  $\delta(1)$  and  $\sigma(1)$ , and place in the buffer store.
- d) Assemble and transfer to the Ballistic Data Store (Block XIII) the six quantities in the buffer store. These replace the oldest entry in the store.

6. Outputs:

6.1 Ballistic Data Store Entry: To Block XIII, a new entry for the data store is transferred. It contains the same six quantities listed in the buffer store (Item 4). The magnitude and accuracies are the same as the same quantities in the Input (Item 3).

6.2 Time of Day Request: To Block XXIX, Real Time Clock, a request for the time of day.

7. Comments: None.

Block XXXI: Prediction Mode Designation (by Operator)

1. Function: To provide the Prediction Module, Block III, with Operator instructions on the types of prediction modes to be used. Operator instructions override the automatic prediction modes.

2. Frequency: Operator's Panel is polled for a change in prediction mode instruction every 0.1 second during the Initialization II Mode and during the Semi-Automatic Fire Control Mode.

3. Inputs:

3.1 Polling Trigger: From Block XXIX, Real Time Clock, the polling trigger is received every 0.1 second for polling the output buffer of the Operator's Panel (Peripheral device D2.a).

3.2 Operator's Panel: From peripheral device D2.a, the change in the activation of one or more prediction function switches is read. These switches are:

- a. USE DIVE:       (i) Activated: Use Operator entered dive angle threshold in Prediction Module (Block III) and deactivate automatic dive and turn thresholds.  
                    (ii) Deactivated: Do not use Operator entered dive angle threshold.
- b. USE LINEAR:   (i) Activated: Use linear prediction and deactivate automatic dive and turn thresholds.  
                    (ii) Deactivated: No instructions to Prediction Module (Block III).
- c. USE TURN:       (i) Activated: Use Operator entered turn acceleration threshold in Prediction Module (Block III) and deactivate automatic dive and turn thresholds.  
                    (ii) Deactivate: Do not use Operator entered turn acceleration threshold.

#### 4. Constants:

4.1 Prediction Mode Status: Status bits indicating which types of prediction modes are to be used.

#### 5. Operations Performed:

- a. Upon receipt of the Polling Trigger, the output buffer of the Operator's Panel (part of Peripheral Device D2.a) is polled for any change in the USE DIVE prediction mode switch setting.
  - (i) If no change, go to Step h.
  - (ii) If activated, go to Step b.
  - (iii) If deactivated, go to Step e.
- b. With activation of the USE DIVE prediction, deactivate both automatic dive and automatic turn predictions in the Prediction Module (Block III). Go to Step c.
- c. Use Stored Value of dive angle threshold from Block XXXII in Prediction Module. Go to Step d.

- d. Light USE DIVE switch on Operator's Panel. Go to Step h.
- e. With deactivation of USE DIVE prediction, reactivate automatic dive and automatic turn predictions in Prediction Module unless blocked by Steps l and/or p. below. Go to Step f.
- f. Use Preset Value of Dive Angle Threshold and Stored Value of Turn (Acceleration) Threshold from Block XXXII in Prediction Module. Go to Step g.
- g. Extinguish USE DIVE switch light on Operator's Panel. Go to Step h.
- h. Poll USE TURN prediction mode switch for any change.
  - (i) If no change, go to Step o.
  - (ii) If activated, go to Step i.
  - (iii) If deactivated, go to Step l.
- i. With activation of USE TURN prediction, deactivate automatic dive and automatic turn predictions in Block III, Prediction Module. Go to Step j.
- j. Use Stored Value of Turn (Acceleration) Threshold from Block XXXII in Prediction Module. Go to Step k.
- k. Light USE TURN switch on Operator's Panel. Go to Step o.
- l. With deactivation of USE TURN prediction, reactivate automatic dive and automatic turn predictions in Prediction Module unless blocked by Steps b and/or p. Go to Step m.
- m. Use Stored Value of Dive Angle Threshold and Preset Value of Turn (Acceleration) Threshold from Block XXXII in Prediction Module. Go to Step n.
- n. Extinguish USE TURN light on Operator's Panel. Go to Step o.
- o. Poll USE LINEAR prediction mode switch for any change.
  - (i) If no change, end processing.
  - (ii) If activated, go to Step p.
  - (iii) If deactivated, go to Step r.
- p. With activation of USE LINEAR prediction, deactivate automatic dive and automatic turn predictions in Prediction Module (Block III). Go to Step q.

- q. Light USE LINEAR switch on Operator's Panel. End processing.
- r. With deactivation of USE LINEAR prediction, reactivate automatic dive and automatic turn predictions in Prediction Module unless blocked by Steps b and/or l. Go to Step s.
- s. Use Stored Values of Dive Angle Threshold and Turn (Acceleration) Threshold from Block XXXII in Prediction Module. Go to Step t.
- t. Extinguish USE LINEAR light on Operator's Panel. End processing.

6. Output:

6.1 Mode Designation: To Blocks III and XXXII, send instructions on the types of linear, dive and constant turn predictions to be used and whether automatic or operator set thresholds are to be used.

7. Comment: None.

Block XXXII: Prediction Mode Thresholds (by Operator)

1. Function: To provide dive angle and turn (acceleration) thresholds for the prediction functions, as inputted by the Operator during Initialization I Mode.

2. Frequency: Operator's Panel is polled for a change in dive angle threshold and turn threshold every 0.1 second during Initialization I Mode.

3. Inputs:

3.1 Polling Trigger: From Block XXIX, Real Time Clock, the polling trigger is received every 0.1 second for polling the output buffer of the Operator's Panel (peripheral device D 1).

3.2 Operator's Panel: From the output buffer of the Operator's Panel, part of peripheral device D 1, the values of the dive angle and turn thresholds are read. Possible values are:

- a. PRESET value trigger for either dive angle or turn thresholds.
- b. Dive Angle Threshold: Operator entered value.  
Magnitude: 2 deg to 15 deg.  
Accuracy: 1 deg.

c. Turn (Acceleration) Threshold: Operator entered value.

Magnitude: 0.2g to 1.0g

Accuracy: 0.1g

3.3 Prediction Mode Designation: From Block XXXI, status word on the type of prediction modes to be used.

4. Constants:

4.1 Preset Values: Value to be used by the Prediction Module (Block III), if (a) Operator enters PRESET or (b) automatic prediction is used.

Dive Angle Threshold: 10 deg absolute value.

Turn (Acceleration) Threshold: 0.3g absolute value.

4.2 Stored Value: Last threshold values entered by the Operator.

d = Dive angle threshold, absolute magnitude.

Magnitude: 2 deg to 15 deg.

Accuracy: 1 deg.

a/g = Turn (acceleration) threshold, absolute magnitude.

Magnitude: 0.2g to 1.0g.

Accuracy: 0.1g.

5. Operations Performed:

Upon receipt of the Polling Trigger during Initialization I Mode, the output buffer of the Operator's Panel (part of Peripheral Device D 1) is polled for changes in the dive angle and turn thresholds.

a. If no change, processing ends.

b. If changed and PRESET has been entered, enter appropriate Preset Value, Item 4.1, in Stored Value. Item 4.2, memory location.

c. If changed and value has been entered, enter this value in the appropriate Stored Value memory location.

The values in the Stored Value memory locations are sent to the Prediction Module, Block III, as instructed by the Prediction Mode Designations of Block XXXI during the Semi-Automatic Fire Control Mode.

6. Outputs: To Block III during Semi-Automatic Fire Control Mode, the appropriate threshold values in Stored Value file (Item 4.2 above). Magnitudes and accuracies as given in Item 4.2.

7. Comment: None

Block XXXIII: Engagement Mode Determination (by Operator)

1. Function: To provide the fire control computer with Operator instructions on the engagement mode to be used. These are: Semi-Automatic Fire Control, Manual, Standby, and Test, and Initialization.
2. Frequency: Operator's Panel is polled for a change in engagement mode instructions every 0.1 second. This interrogation continues in all operating modes.
3. Inputs:
  - 3.1 Polling Trigger: From Block XXIX, Real Time Clock, the polling trigger is received every 0.1 second for polling the output buffer of the Operator's Panel (Peripheral Devices D 2.a and D 8.a) for operating mode instruction changes.
  - 3.2 Operator's Panel: From peripheral devices D 2.a and D 8.2, the change in the activation of one of the engagement mode switches is read. When activated, they mean:
    - (a) SCAN: AFAADS enters the Semi-Automatic Fire Control Mode.
    - (b) MANUAL: AFAADS is in the Manual Fire Control Mode. Computer is not used in this mode.
    - (c) BEGIN NEW TARGET: Breakoff current engagement and search for a new target. AFAADS remains in the Semi-Automatic Fire Control Mode.
    - (d) BEGIN INIT I  
(Begin Initialization I): Computer enter the Initialization I Mode for the purpose of operator entry of Initialization I and II data.
    - (e) STANDBY II: Computer goes to the Standby Mode with initialization data.
    - (f) BEGIN TEST: Computer goes to the Test Mode.
4. Constants: Engagement mode status bits, indicating engagement mode being used.
5. Operations Performed: Upon receipt of the Polling Trigger, the appropriate elements of the output buffer of the Operator's Panel (parts of peripheral devices D 2.a and D 8.a) are polled for any changes. If any change is detected, the computer and entire fire control system is switched to the indicated new operating mode. Actions include:

- a. Initiate target acquisition for a Semi-Automatic Fire Control Mode.
- b. Go to Manual Mode, with the computer in Standby II.
- c. Go to Standby Mode II, clearing the tracking filter.
- d. Go to test mode.
- e. Enter the Initialization I mode.
6. Outputs:
  - 6.1 Operating Mode Status: To Blocks II, III, IV, and VI, send the change in the operating mode for AFAADS.
  - 6.2 Operator's Panel: To the Operator's Panel send status signals to light/extinguish the appropriate indicators.
7. Comments: None

Block XXXIV: Projectile Type Selection (by Operator)

1. Function: To provide the Ballistic Computation Module (Block IV), the Muzzle Velocity Store (Block IX), and the two bias correction stores (Blocks XXII and XXXVII) with the type of projectile being fired.
2. Frequency: Operator's Panel is polled for a change in the type of projectile to be fired every 0.1 second during the Initialization Mode.
3. Inputs:
  - 3.1 Polling Trigger: From Block XXIX, Real Time Clock, the polling trigger is received every 0.1 second during the Initialization Mode for polling the output buffer of the Operator's Panel (Peripheral Device D 7) for changes in projectile type.
  - 3.2 Operator's Panel: From Peripheral Device D 7 (part of the Operator's Panel), any changes in the activation of one of the five projectile type switches.
4. Constants:
  - 4.1 Projectile Type File: Chosen projectile type, one of five.
5. Operations Performed:

During the Initialization Mode, and upon receipt of the Polling Trigger, the five projectile type switches are polled for any changes in activation.

- (a) If no change, end processing.
- (b) If change, change value in projectile type file and change lighting on Operator's Panel.

A change in the projectile type will cause:

- (a) Recomputation of the ballistic coefficients in the ballistic equation - see Block IV.
- (b) Selection of different factors for azimuth, elevation, and velocity bias correction - see Blocks XXII and XXXVII.

6. Outputs:

6.1 Projectile Type: To Blocks IV, IX, XXII, XXIV, and XXXVII, send status word indicating the projectile type being used.

6.2 Operator's Panel: To the Operator's Panel send appropriate signals to light and extinguish the appropriate projectile type indicators.

7. Comments: None

Block XXXV: Coordinate Conversion (Earth to Vehicle)

1. Function: To convert the corrected gun angles from polar earth coordinates to vehicle coordinates.

2. Frequency: Whenever gun orders are generated, i. e., once every 0.1 second.

3. Inputs:

3.1 Corrected Gun Angles in Earth Coordinates: From Block V.

A(goj) = Ballistically corrected gun azimuth angle in earth coordinates; measured clockwise from front of vehicle.

Magnitude: 0 deg to 360 deg.

Accuracy: \_\_\_\_\_ mils.

E(goj) = Ballistically corrected gun elevation angle in earth coordinates; positive up.

Magnitude: 0 deg to 85 deg.

Accuracy: \_\_\_\_\_ mils.

3.2 Vehicle Orientation: From Vehicle Orientation Sensors, peripheral device D 7,

$\theta(v)$  = Pitch or lengthwise tilt of the vehicle relative to the surface of the earth; positive for front up.

Magnitude: -35 deg to +35 deg.

Accuracy: 0.6 deg.

$\phi(v)$  = Cant or crosswise tilt of the vehicle about its axis; positive clockwise or right side down. (Note: Cant is executed after pitch).

Magnitude: -35 deg to +35 deg.

Accuracy: 0.6 deg.

4. Constants: None.

5. Operations Performed:

5.1 Polar Earth Coordinates to Cartesian Earth Coordinates

Let  $X(goj)$ ,  $Y(goj)$ , and  $Z(goj)$  be the earth oriented relative Cartesian coordinates (i. e., the Cartesian coordinates divided by target range).

Then

$$X(goj) = \cos E(goj) \sin A(goj)$$

$$Y(goj) = \cos E(goj) \cos A(goj)$$

$$Z(goj) = \sin E(goj)$$

5.2 Cartesian Earth Coordinates to Cartesian Vehicle Coordinates

Let  $X(gj)$ ,  $Y(gj)$ , and  $Z(gj)$  be the vehicle oriented relative Cartesian coordinates (i. e., Cartesian coordinates divided by target range). Then

$$X(gj) = X(goj) \cos \phi(v) + Y(goj) \sin \theta(v) \sin \phi(v) - Z(goj) \cos \theta(v) \sin \phi(v)$$

$$Y(gj) = Y(goj) \cos \theta(v) + Z(goj) \sin \theta(v)$$

$$Z(gj) = X(goj) \sin \phi(v) - Y(goj) \sin \theta(v) \cos \phi(v) + Z(goj) \cos \theta(v) \cos \phi(v).$$

5.3 Cartesian Vehicle Coordinates to Polar Vehicle Coordinates:

Let  $E(gj)$  and  $A(gj)$  be the vehicle oriented gun elevation and azimuth pointing angles.

$$E(gj) = \sin^{-1} Z(gj)$$

$$A(gj) = \begin{cases} \sin^{-1} [X(gj)/\cos E(gj)] \\ \text{or} \\ \cos^{-1} [Y(gj)/\cos E(gj)] \end{cases}$$

6. Outputs:

6.1 Corrected Gun Angles in Vehicle Coordinates: To Block XXXVI

$A(gj)$  = Ballistically corrected gun azimuth angle in vehicle coordinates; measured clockwise from front of vehicle.

Magnitude: 0 deg to 360 deg.

Accuracy: \_\_\_\_\_ mils.

$E(gj)$  = Ballistically corrected gun elevation angle in vehicle coordinates; positive up.

Magnitude: -5 deg to +85 deg.

Accuracy: \_\_\_\_\_ mils.

7. Comments: None.

Block XXXVI: Gun Lead Angles

1. Function: To compute the gun lead angles relative to the sensor target angles. The gun and sensor are on the same mount with the gun traveling relative to the sensor.

2. Frequency: Whenever gun orders are generated, i. e., every 0.1 second.

3. Inputs:

3.1 Corrected Gun Angles: From Block XXXV.

$A(gj)$  = Ballistically corrected gun azimuth angle in vehicle coordinates; measured clockwise from front of vehicle.

Magnitude: 0 deg to 360 deg.

Accuracy: \_\_\_\_\_ mils.

$E(gj)$  = Ballistically corrected gun elevation angle in vehicle coordinates; positive up.

Magnitude: -5 deg to +85 deg.

Accuracy: \_\_\_\_\_ mils.

3.2 Target Position: From Block XX.

$\underline{A}(mj)$  = Current target azimuth at time  $t(j)$ , measured clockwise from front of vehicle.

Magnitude: 0 deg to 360 deg.

Accuracy: \_\_\_\_\_ mils.

$\underline{E}(mj)$  = Current target elevation at time  $t(j)$ , positive up.

Magnitude: -5 deg to +85 deg.

Accuracy: \_\_\_\_\_ mils.

4. Constants: None

5. Operations Performed: Let  $\delta(bcj)$  and  $\sigma(bcj)$  be the azimuth and elevation gun lead angles relative to the sensor.

$$\delta(bcj) = A(gj) - \underline{A}(mj)$$

$$\sigma(bcj) = E(gj) - \underline{E}(mj)$$

6. Outputs:

6.1 Gun Lead Angles: To Block XXII

$\delta(bcj)$  = Ballistically corrected azimuth gun lead angle relative to target; positive to right of target.

Magnitude: -90 deg to +90 deg.

Accuracy: 0.25 mil.

$\sigma(bcj)$  = Ballistically corrected elevation gun lead angle relative to target; positive above target.

Magnitude: -30 deg to +70 deg.

Accuracy: 0.25 mil.

7. Comment: None

Block XXXVII: Azimuth and Elevation Bias Correction

1. Function: To apply the azimuth and elevation bias correction factors to the gun's azimuth and elevation lead angles. These correction factors are assumed to be a function of the ammunition type and the azimuth and elevation of the gun and are updated by the projectile miss distance measurements.

2. Frequency: Every update of the gun train and elevation orders, i. e., every 0.1 second.

3. Inputs:

3.1 Angle Bias Correction Factors: From Block XVI.

$A(bj)$  = Azimuth angle bias correction factor to be applied to the gun azimuth lead angle, positive clockwise.

Magnitude: -2 deg. to +2 deg.

Accuracy: 0.1%.

$E(bj)$  = Elevation angle bias correction factor to be applied to the gun elevation lead angle, positive up.

Magnitude: -2 deg. to +2 deg.

Accuracy: 0.1%.

Note: The algebraic sign of these bias correction factors will, when added to the gun lead angles, remove the biases.

3.2 Ammunition Selection: From Block XXXIV, one of five selected ammunition types.

3.3 Velocity Corrected Lead Angles: From Block XXII

$\delta(vcj)$  = Muzzle velocity corrected gun lead angle in azimuth; positive clockwise.

Magnitude: -90 deg. to +90 deg.

Accuracy: \_\_\_\_\_ mils.

$\sigma(vcj)$  = Muzzle velocity corrected gun lead angle in elevation; positive up.

Magnitude: -30 deg. to +70 deg.

Accuracy: \_\_\_\_\_ mils.

3.4 Target Direction: From Block XX

$A(mj)$  = Current target azimuth angle at time  $t(j)$ , measured clockwise from front of vehicle.

Magnitude: 0 deg. to 360 deg.

Accuracy: \_\_\_\_\_ mils.

$E(mj)$  = Current target elevation angle at time  $t(j)$ , positive up.

Magnitude: -5 deg. to +85 deg.

Accuracy: \_\_\_\_\_ mils.

#### 4. Constants:

4.1 Lead Angle Corrections: Store the last used values of the gun lead angle corrections in azimuth and elevation due to angle bias. Storage is by ammunition type (5 types), by gun azimuth (every  $30^\circ$  in  $360^\circ$  or 12 locations), and by gun elevation (-5 deg. to +10 deg., 10 deg. to 30 deg., above 30 deg. or 3 locations). (Total of 180 files).

#### 5. Operations Performed:

Logic flow is illustrated in Figure 11-11.

- a) Gun Pointing Direction: On every update of the gun train and elevation orders, determine the azimuth and elevation of the gun relative to the AFAADS vehicle.

$$A(gj) = \underline{A}(mj) + \delta(vcj)$$

$$E(gj) = \underline{E}(mj) + \sigma(vcj)$$

Continue with Step b.

- b) Memory Indexing: Using (1) the gun azimuth and elevation angles as determined in Step a and (2) the selected ammunition types (Input Item 3.2), determine the address of the appropriate gun lead angle corrections stored in memory (Item 4.1). Continue with Step c.
- c) Correction Logic Determination: Determine if angle bias correction factors are available from Block XVI (Input Item 3.1). If no factors are available, proceed to Step d; if the factors are available, proceed to Step e.
- d) Data Retrieval: If no new correction factor data are available, retrieve from memory (Item 4.1) the last used gun lead angle correction factors in azimuth and elevation for the ammunition type and the azimuth and elevation bins determined in Step b. Then proceed to Step f.
- e) Store: If new correction factor data are available and for the selected memory locations determined in Step b, replace in memory (Item 4.1) the old gun lead angle correction factors in azimuth and elevation by the angle bias correction factors available from Block XVI. Proceed to Step f.

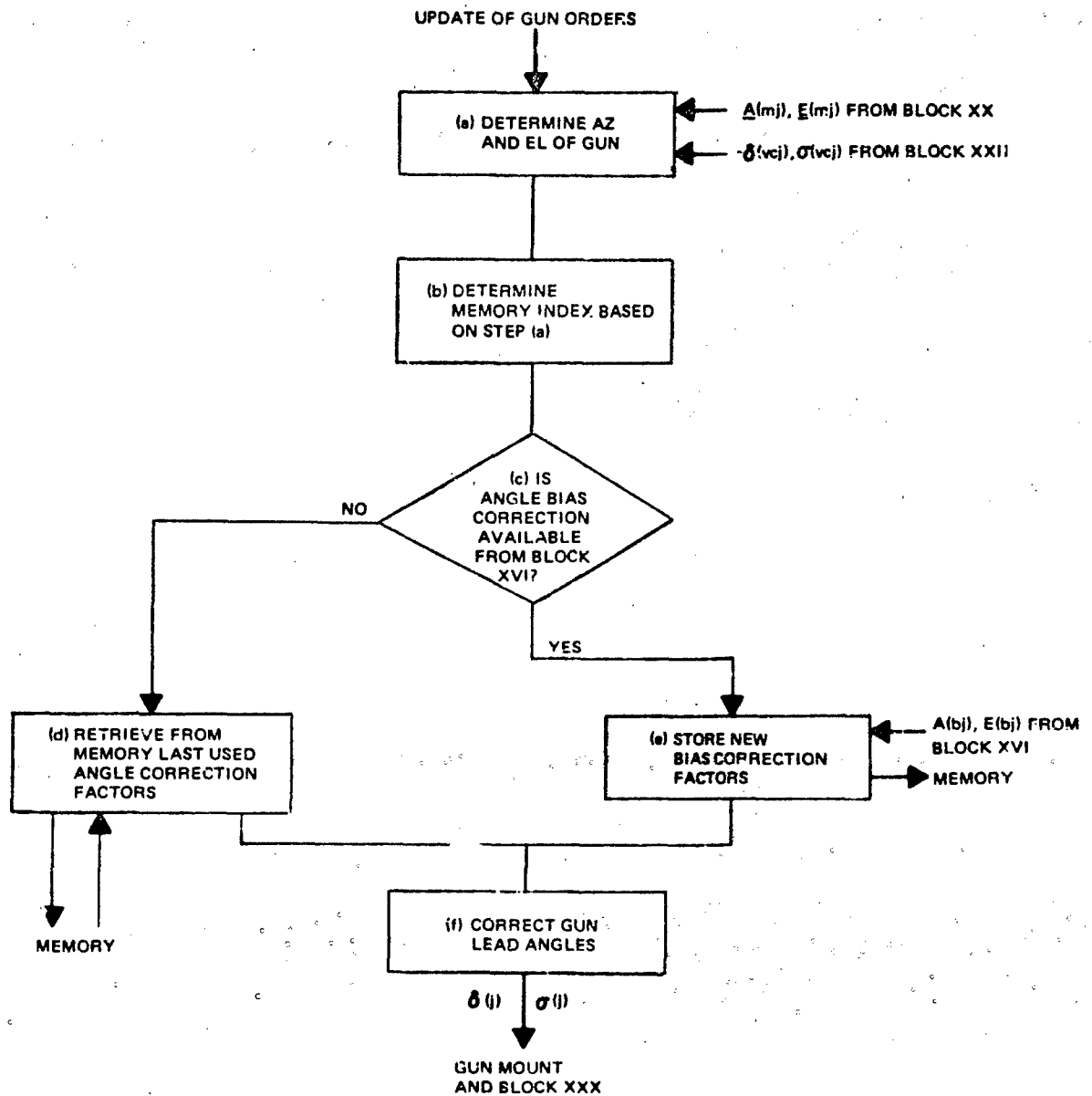


Figure 11-11. Angular Bias Correction Logic.

- f) Lead Angle Corrections: Apply the gun lead angle correction factors (from memory in Step d or new input from Step c) to the muzzle velocity corrected gun lead angles.

Azimuth  $\delta(j) = \delta(vcj) + A(bj)$

Elevation  $\sigma(j) = \sigma(vcj) + E(bj)$

6. Outputs:

6.1 Corrected Gun Lead Angles: To Gun Mount (Peripheral Device D 5) and to Block XXX.

$\delta(j)$  = Fully corrected gun lead angle in azimuth; positive clockwise.  
Magnitude: -90 deg. to +90 deg.  
Accuracy: 0.25 mil.

$\sigma(j)$  = Fully corrected gun lead angle in elevation; positive up.  
Magnitude: -30 deg. to +70 deg.  
Accuracy: 0.25 mil.

7. Comments: None

Block XXXVIII: Residual Miss

1. Function: To determine the residual miss angles caused by the azimuth, elevation, and muzzle velocity biases. This miss is the difference between the measured closed loop miss distance and that caused by target maneuvers and other target track prediction errors.

2. Frequency: Whenever a valid projectile miss distance measurement is made, no more often than every 0.1 second.

3. Inputs:

3.1 Transverse Target Maneuver Correction: From Block XXIII.

$\Delta T(vj)$  = Transverse target maneuver correction at time  $t(j)$ , or the correction due to target maneuver in a plane perpendicular to the elevation plane and through the line-of-sight to the target, positive for target to right of predicted position.  
Magnitude: -45 deg to +45 deg.  
Accuracy: \_\_\_\_\_ mils.

3.2 Elevation Target Maneuver Correction: From Block XXVIII

$\Delta E(vj)$  = Elevation miss angle due to target maneuver at time  $t(j)$ ,  
positive for the target above the predicted position.  
Magnitude: -30 deg to +30 deg.  
Accuracy: \_\_\_\_\_ mils.

3.3 Measured Projectile Miss Distance: From Block XXIV

$\Delta T(mj)$  = Valid measured transverse projectile miss angle at the equal  
range time  $t(j)$ . This is the measured miss distance through  
the line-of-sight to the target and perpendicular to the  
elevation miss distance. Positive if miss is to the right.  
Magnitude: -4 deg to -0.008 deg (0.14 mil) and  
+0.008 deg (0.14 mil) to +4 deg.  
Accuracy: 0.25 mil.

$\Delta E(mj)$  = Valid measured elevation projectile miss angle at the equal  
range time  $t(j)$ . Positive if the miss is up.  
Magnitude: -2 deg to -0.008 deg (0.14 mil) and  
+0.008 deg (0.14 mil) to +2 deg.  
Accuracy: 0.25 mil.

4. Constants: None

5. Operations Performed: Using the geometric relationships for the first  
defined angles as illustrated in Figure 11-12, the residual miss angles are:

$$\Delta T(j) = \Delta T(mj) + \Delta T(vj)$$

$$\Delta E(j) = \Delta E(mj) + \Delta E(vj)$$

6. Outputs:

6.1 Residual Miss Vector: To Block XVI

$\Delta T(j)$  = Transverse residual miss angle at time  $t(j)$ , in a plane  
through the line-of-sight and perpendicular to the elevation  
plane. Positive for miss to right of predicted target position.  
Magnitude: -2 deg to +2 deg.  
Accuracy: 0.5 mil.

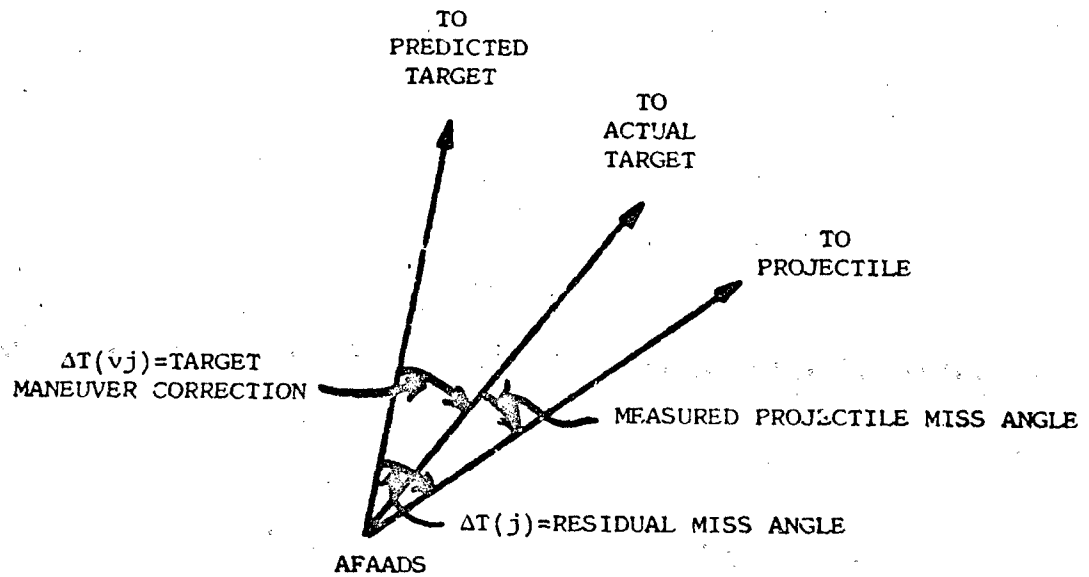


Figure 11-12. Definition of Transverse Miss Angles

$\Delta E(j)$  = Elevation residual miss angle at time  $t(j)$ , positive for miss above predicted target position.

Magnitude: -2 deg to +2 deg.

Accuracy: 0.5 mil.

7. Comment: The angle definitions are consistent in this and related blocks.

#### Block XXXIX: Time of Projectile Fire

1. Function: To compute the time of day when the projectile would have had to be fired in order to impact at current time  $t(j)$ .

2. Frequency: Whenever a valid projectile miss distance measurement is made, no more frequent than every 0.1 second.

3. Inputs:

3.1 Projectile Time of Flight: From Block XII

$t(poj)$  = Computed projectile time of flight required to impact a target at its current position, i. e., at time  $t(j)$ .

Magnitude: 0 to 10 seconds.

Accuracy: 0.005 second.

3.2 Time-of-Day: From Block XXIV

$t(j)$  = Time of day when the valid miss angles were measured.

Magnitude: 0 to about 5 minutes.

Accuracy: 0.005 second.

4. Constants: None

5. Operations Performed: Time when the projectile must have been fired to impact at time  $t(j)$  is

$$t(j-p) = t(j) - t(poj)$$

6. Outputs:

6.1 Projectile Time of Fire: To Block XIV

$t(j-p)$  = Time of day when projectile was fired.

Magnitude: 0 to about 5 minutes.

Accuracy: 0.005 second.

7. Comments: In forming the time difference, a negative result must be interpreted as lying in the previous time-of-day time slot, and hence the result is both positive and slightly less than the maximum time-of-day value.

Block XL: Coordinate Conversion (Vehicle to Earth)

1. Function: To convert the predicted target position in Cartesian vehicle coordinates to Earth oriented polar coordinates. This provides the correct coordinate data for the ballistic corrections of Block V.

2. Frequency: Whenever gun orders are generated, i. e., every 0.1 second.

3. Inputs:

3.1 Predicted Target Position: From Block IV the predicted target position for a convergent ballistic solution.

$X(pj)$  = Predicted cross position of target relative to AFAADS at time of projectile impact; positive on right.

Magnitude: -10,000m to +10,000m.

Accuracy: \_\_\_\_\_m.

$Y(pj)$  = Predicted lengthwise position of target relative to AFAADS at time of projectile impact; positive forward.

Magnitude: -10,000m to +10,000m.

Accuracy: \_\_\_\_\_m.

$Z(pj)$  = Predicted target elevation relative to AFAADS at time of projectile impact, positive up.

Magnitude: -900m to +8,000m.

Accuracy: \_\_\_\_\_m.

$D(pj)$  = Predicted range to the target at time of projectile impact.

Magnitude: 100m to 10,000m.

Accuracy: \_\_\_\_\_m.

3.2 Vehicle Orientation: From Vehicle Orientation Sensor Input, peripheral device D 7.

$\theta(v)$  = Pitch or lengthwise tilt of the vehicle relative to the surface of the earth; positive for front up.

Magnitude: -35 deg to +35 deg.

Accuracy: 0.6 deg.

$\phi(v)$  = Cant or crosswise tilt of the vehicle about the vehicle axis; positive for clockwise rotation or right side down. (In going from a horizontal position, the vehicle first pitches up, and then cants.)

Magnitude: -35 deg to +35 deg.

Accuracy: 0.6 deg.

4. Constants: None

5. Operations Performed:

5.1 Cartesian Vehicle Coordinates to Cartesian Earth Coordinates:

Let  $X(poj)$ ,  $Y(poj)$ , and  $Z(poj)$  be the earth oriented Cartesian coordinates.

$$X(poj) = X(pj) \cos \phi(v) + Z(pj) \sin \phi(v)$$

$$Y(poj) = X(pj) \sin \theta(v) \sin \phi(v) + Y(pj) \cos \theta(v) - Z(pj) \sin \theta(v) \cos \phi(v)$$

$$Z(poj) = X(pj) \cos \theta(v) \sin \phi(v) + Y(pj) \sin \theta(v) + Z(pj) \cos \theta(v) \cos \phi(v)$$

5.2 Cartesian Earth Coordinates to Polar Earth Coordinates

Let  $E(oj)$  and  $A(oj)$  be predicted target elevation and azimuth in earth coordinates.

$$E(oj) = \sin^{-1} [Z(poj)/D(pj)]$$

$$A(oj) = \begin{cases} \sin^{-1} [X(poj)/D(pj) \cos E(oj)] \\ \text{or} \\ \cos^{-1} [Y(poj)/D(pj) \cos E(oj)] \end{cases}$$

6. Outputs:

6.1 Predicted Target Position in Polar Earth Coordinates: To Block V.

A(oj) = Predicted target azimuth relative to the surface of the earth at the time of projectile impact, measured clockwise from the front of the vehicle.

Magnitude: 0 deg to 360 deg.

Accuracy: \_\_\_\_\_ mils.

E(oj) = Predicted target elevation relative to the surface of the earth at the time of projectile impact, positive up.

Magnitude: 0 deg to 85 deg.

Accuracy: \_\_\_\_\_ mils.

7. Comment: None

## SECTION 12

### DEGRADED MODES

#### 12.1 INTRODUCTION

In any system some malfunctions can be expected. These can be due to equipment failures or partial battle damage. Thus, it is necessary that the designers anticipate problems and plan emergency operating plans, so as to maintain at least a degraded mode capability.

Knowing this, the computer concepts task has included a preliminary analysis of the effect of specific equipment failures upon system operation. These have been divided into non-critical or generally temporary failures and critical ones. The following two subsections deal with these topics.

#### 12.2 NON-CRITICAL FAILURE

A very basic requirement of any system is that it be able to recognize and deal with the simpler failures. Assume a well engineered AFAADS gun system. We can expect to have equipment that malfunctions rarely. However, we must address the problems and limitations in state-of-the-art equipment. It is not unreasonable to expect an occasional misfire, or a noisy target signal. These "failures" are termed "non-critical". Still, they could be major problems unless planned for. Some are discussed.

##### 12.2.1 Types of Non-Critical Failures

Four non-critical "failures" that will not cause interrupts in system use are:

- 1) Angular rates of the sensor too high, the slant range (laser-measured) lost for a few scans, or an occasional "very noisy" signal received.
- 2) Target goes behind a terrain obstacle causing loss of angle and range data, so no signal returns are obtained. This causes the target tracking filter to receive artificially -

generated target position data.

- 3) FLIR and/or laser malfunctions for a period of up to two seconds.
- 4) There is no target. (This is the case during a test, when a target is simulated for a period of up to sixteen seconds.)

Note that gyros and gun servos are expected to always function. Because these elements are external to the computer, they are assumed to have their own separate alarms.

#### 12.2.2 Planned Corrective Action

Since the above problems can be expected to occur, at least occasionally, under operational conditions; degraded mode AFAADS capabilities have been designed to meet the problems associated with each. The planned corrective actions discussed below correspond to the types of malfunctions discussed in Subsection 12.2.1 above.

- 1) If the angular rate of the target becomes very high, the angle tracking servo may only be able to follow with a large lag angle. This large lag angle may become sufficient that the target is no longer within the very narrow field of view of the laser (an order of magnitude less than the FLIR). The Operator would then receive notice of "Range Data Lost". It is anticipated that this condition will last for only a few scans, when the target angular rate drops and the angle servo can catch up.

For data that is noisy, nothing need be done. The smoothing filter will largely eliminate ordinary noise. Very noisy data may cause the sensor servos to "jump". This jump might cause the laser to miss the target for a couple of scans. Ordinarily, however, the laser and FLIR will follow the target without problems. The "Range Data Lost" indicator would light, notifying the gunner that range data has been lost.

- 2) Should the target go behind an obstacle, the simultaneous loss of both angle and range data would cause the computer to indicate it is "Regenerating Position". According to the AFAADS Simulation Model, computer generation of position will work for a time, and that once the target is again in view, the sensors will find the target, with no loss in tracking.

However, should the sensors not redetect the target after a sufficient time (set at two seconds), either because it is still behind an obstacle or because of a drastic maneuver by the aircraft while it is behind the obstacle, ERROR LOST TARGET will be indicated, showing that two seconds (or twenty-one consecutive target position points) have elapsed with no laser or FLIR data over that period. If the target is in acquisition sensor view, the gunner-operator knows the sensors have lost the target and he must re-acquire the target. He presses BEGIN NEW TARGET, "SCAN" lights, and he must lock on the target again.

If the two seconds have elapsed and the target is still not in view, he has two options. If the target seems to have left the area, he can cause the system to stop and wait for a new engagement by pressing STANDBY II. Otherwise, he may allow the system to remain in the target regeneration mode, hoping the sensors will reacquire target once it again comes into sensor view.

- 3) If the laser and/or FLIR do not fire, the system will assume it has missed a point, regenerate that point, and maintain operations. If only laser data is missed, it will indicate RANGE DATA LOST. If only FLIR data is missed, it will indicate ANGLE DATA LOST. If both are missed simultaneously (unlikely), REGENERATING POSITION will be indicated, and operations will proceed as if target were behind an obstacle.

- 4) Perhaps a test is not properly a degraded mode, but the logic used if there is no target (i.e., a test is taking place) is almost identical to the logic used for "fixing up" if no target signals are obtained. The computer, once it begins "Testing", will initiate proper logic, so the entire instruction sequence is tested via a dynamic system test. Obviously there is no target, but the computer must order the sensors to slew at the proper rates, so projectiles fired during test will be sensed. Therefore, the "target position" is continuously computed and indicated by the REGENERATING POSITION indicator. Logic insures ERROR LOST TARGET indicator will not light, even though a whole test is sixteen seconds in duration.

### 12.3 CRITICAL FAILURES

Short-term (under two second) sensor malfunctions are anticipated. More important failures might render the system useless. Still others could force the Operator to revert to the Manual Mode, which is certainly a degraded mode. These long-term failures all have the effect of seriously degrading or even terminating use of AFAADS.

#### 12.3.1 Types of Critical Failures

Among these serious failures, the most critical include:

- 1) Computer stops working. An "Alarm" indication would be given in that case. The present sizing includes automatic self-tests and an Operator ordered confidence test early in the Test Mode to indicate this failure.
- 2) Laser stops working. With the FLIR still working, the system simulates the targets position; but with one coordinate in target position, missing fire would grow increasingly inaccurate.
- 3) FLIR stops working. For a period, the laser might be able to sense the target; eventually, however, since no angular error signals were fed to the sensor mount, the sensors would

certainly lose the target.

- 4) FLIR and laser not correctly aligned. During a test, if consistent laser misses were noted, possibility of this non-alignment would be indicated. This could be an extension to the indicated results of the Dynamic System Test. During an engagement, the effect would be similar to 2) above, causing a critical degradation of system performance. Any mis-alignment of over a few mils would be indicated by "RANGE DATA LOST" indication.
- 5) Sensor gyros or gun servo stops working. The present logic is executed in such a way that the situation would cause the system to not function. The gun servos are expected to accept position data from the computer; should they fail to do so, the hardware associated with the gun mount is expected to have an alarm outside the computer system designating that failure. The gyro system is assumed to be designed with a similar capability.

#### 12.3.2 Alternatives

There are some alternatives to these and other critical failures. Degraded mode solutions are now discussed. The first five correspond to the above five types of critical failures. The last two deal with operational rather than equipment degradations.

- 1) Should the computer cease to process, the Gunner may revert to manual use of the system (the Manual Mode), taking the computer, "out of the loop".
- 2) Laser failure causes the system to lose one of three coordinates for tracking. The Gunner may use the system manually if he desires to do so.
- 3) FLIR failure would probably require a manual use of the system.
- 4) Alignment problems in the FLIR/laser sensor system could be

detected by the computer, but not easily. Detection of these errors is assumed to be a Gunner responsibility. Manual Mode operation would be necessary until realignment is obtained.

- 5) Sensor gyros and gun servo alarms are assumed to be separate components of the system.
- 6) System becomes "dangerous" to use. Among possible dangers are:
  - a) Acquisition and firing on a "friendly" aircraft. All IFF or plane identification is done external to the computer system, either by separate IFF equipment or visually by the gunner.
  - b) Gunfire into friendly area. The Oerlikon type ammunition will self-destruct in eleven seconds from the time of firing, but no safety routines are in the logic at this time. Most would be non-standard, in any case.
  - c) Misfire by weapon. All weapon fire is assumed to be outside of computer control; no monitoring of fire or gun safety measures are included in this sizing task.
  - d) Overheating, overstressing, etc., of any component of the system. The Computer Self-Test and the Operator ordered Confidence Test will indirectly show electronic/thermal failures by indicating "blown" components through its alarm system; with quality components, this kind of failure would be unlikely. Far more likely problems are stress induced by the angular rates and accelerations of the gun caused by closing, high-speed aircraft; these loads are heavy.
- 7) System danger indications may be separated as follows:
  - a) Firing on an aircraft is done by the Gunner. Hence, he is given responsibility for aircraft identification.

- b) Not firing into a friendly area is the Gunner's responsibility also.
- c) Since firing is done outside computer, the Gunner or gun hardware should detect gun firing problems.
- d) The computer does its own error checking. Separate hardware, outside the computer, should detect other system failures.

## SECTION 13

### TEST MODE

It is desirable that any system have a way of testing itself or have the ability to be tested from an external source. The AFAADS gun-fire system has no planned external links when in the field except for voice radio communications. Thus, the system must have the capability for self-testing.

#### 13.1 Test Concepts Investigated

Under this computer concepts task, investigations on self-testing of AFAADS in the field have been confined to:

- 1) The computer itself and its interfaces with the sensors, gun, Operator's panel, etc.
- 2) System dynamic testing of the complete AFAADS gun weapon system.

Thus, field testing concepts for the separate elements of sensors, gun, servos, etc., have been considered as outside the scope of this work. Obviously, they are important to the overall AFAADS air defense capability; but the detection of critical failures in these elements has not been investigated. Section 12, Degraded Modes, does, however, discuss how failures in certain of these elements can be "fixed-up" to provide a degraded air defense capability.

The self-test concepts that have been investigated are in two portions or phases. Phase one is the "Confidence Test", a complete computer self-test. It consists of the standard tests of computer memory and, perhaps, of the arithmetic and logic. Phase two is a system dynamic test. It is in two parts, the initialization section and the System Dynamic Test itself. In this test, the entire system is tested via the simulation of a target and monitoring of system reaction, by both the computer and the operator. The system is "run through its paces", if you will, and the interaction of the gun/computer/sensor combination is

monitored. Any requirements for adjustments and/or repairs are indicated.

The following discussion is not meant to be a test manual for AFAADS. Rather, the presentation provides some depth to this study contract, detailing the operation of the system throughout a test. The following pages provide an outline and example of what this effort has produced. The implementation of an AFAADS test may have to be done some other way; still, one can see what has been done, what has been planned, and what credibility should be given to these AFAADS design concepts. From this discussion, a planner can establish a meaningful tradeoff of computer costs (monetary, complexity, etc.) of testing versus improved operational capabilities. The Typical Fire Mission presented in Subsection 5.2 provided a cursory scenario of a typical semi-automatic fire control mode engagement. The following paragraphs do the same for the Test Mode.

### 13.2 Typical Self-Test Sequence

A logical time to perform a self-test is when AFAADS first reaches an assigned defense position or in the morning after the system has been warmed up. The operational sequence has numerous similarities to that describing a typical semi-automatic fire mission (see Subsections 5.2, 7.2, and 7.3). To aid in an understanding of the specific operator actions, reference should also be made to the Operator's Panel, Figure 5-6.

#### 13.2.1 Initialization Procedures

After the AFAADS has been turned-on and warmed-up, STBY I (Stand-by I) indicator lights on the Operator's Panel. This shows Initialization I and II Mode parameters can be entered. The procedure is identical to that used before a fire mission except the projectile type selection will probably be "Test". (Reference is made to Subsection 7.2 for a detailed discussion.)

#### 13.2.2 Computer Confidence Self-Test

The remainder of this discussion details the procedures unique to the Test Mode. A top level logic flow diagram is given in Figure 13-1;

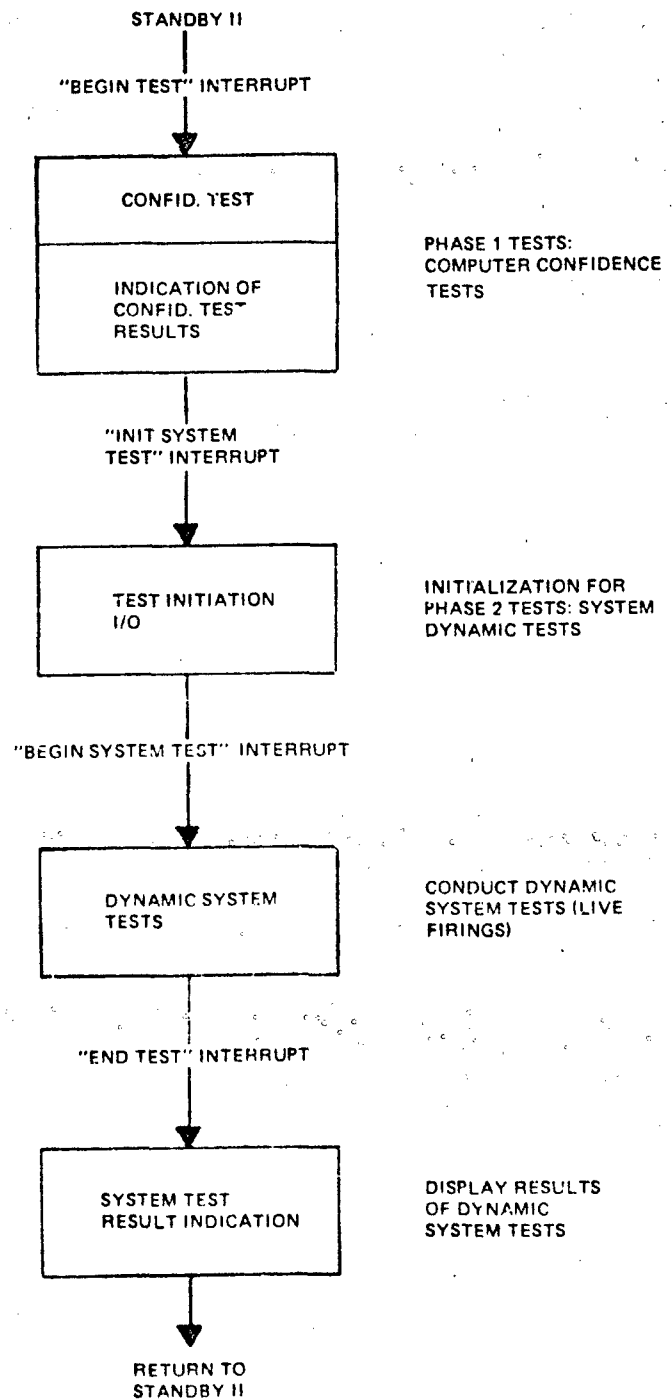


Figure 13-1. Top Level Test Logic Flow Diagram.

greater detail in Figure 13-2. Reference should be made to these figures for the software logic associated with the following operational description.

Phase one of the Test Mode is the computer confidence self-test and it begins with activation of the BEGIN TEST button. Immediately, the CONFID TEST (Confidence Test) indicator lights. For a time, extensive tests by the computer are run. The ordinary self-testing procedures discussed in Sections 9 and 10 will certainly be included. (These are "Check-Sum" operations, where the computer memory is tested.) Other, more exotic, tests may be added but further self-test procedures are not included at present.

Finally, once the operator/gunner is satisfied that the computer is working adequately, i.e., ALARM does not start to flash; he will enter Phase II of the test procedure.

#### 13.2.3 System Dynamic Test

Phase two of the testing or the system dynamic test is basically an input-output operation of the entire AFAADS gun system. The computer generates a simulated target. The sensors are slaved to this target through the regeneration algorithms. A ballistic solution is generated and the gun slews to the indicated azimuth and elevation. The operator opens fire. Projectile miss distance measurements are made, i.e., the "miss" between the simulated target and the actual projectiles. Bias corrections are generated and applied to the gun orders. The results are also recorded for later display.

Let us consider a specific test situation. Figure 13-3 shows an AFAADS vehicle on a low knoll. For test purposes it is assumed to be defending a point target 200 meters due north of it. However, the deployment of other friendly units in the area only permits test firings over a portion of the northeast quadrant, as measured from the fire unit (see Figure 13-4). Thus the dynamic system tests can be conducted against only a portion of the simulated target track shown in Figure 13-4. Figure 13-5 amplifies the coordinate system used.

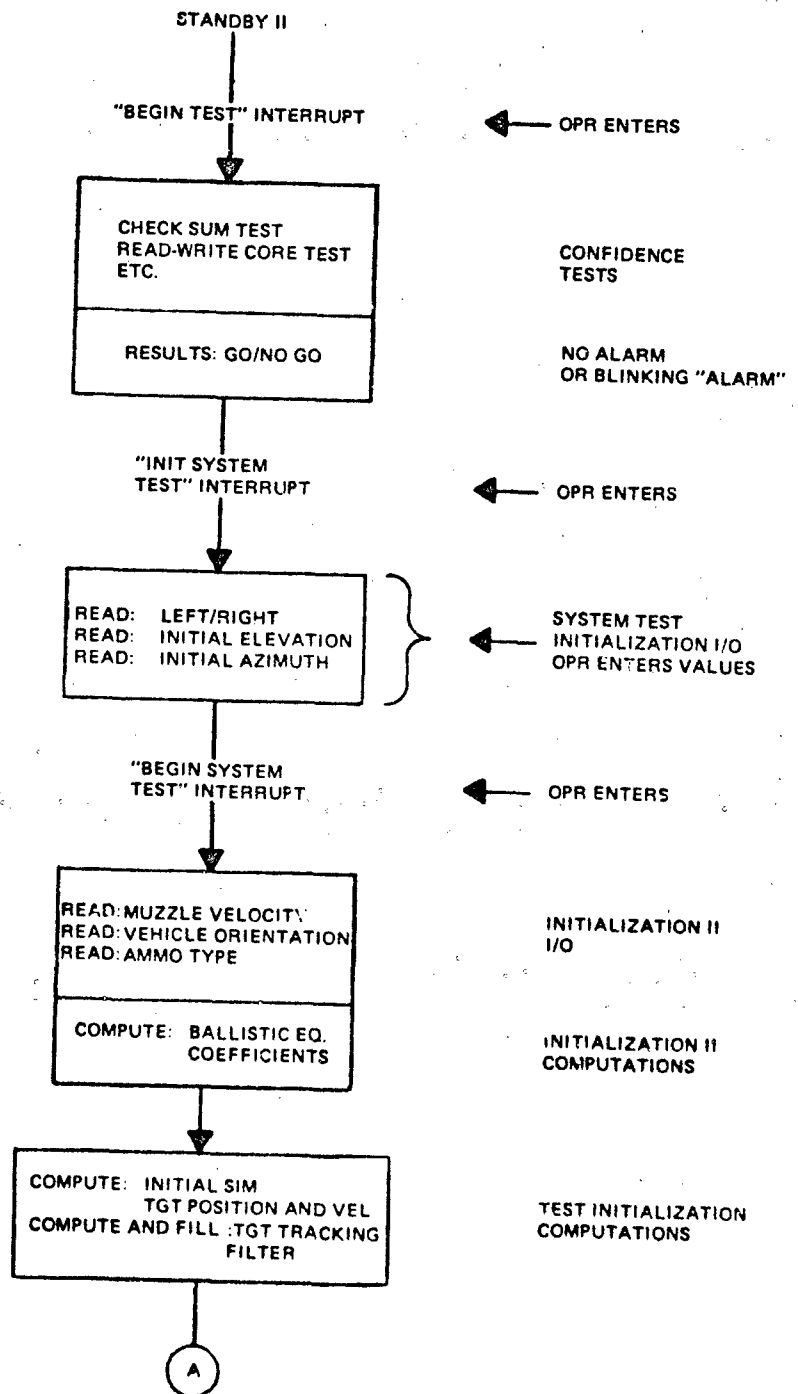


Figure 13-2. Medium Level Test Logic Flow Diagram (Sheet 1 of 2).

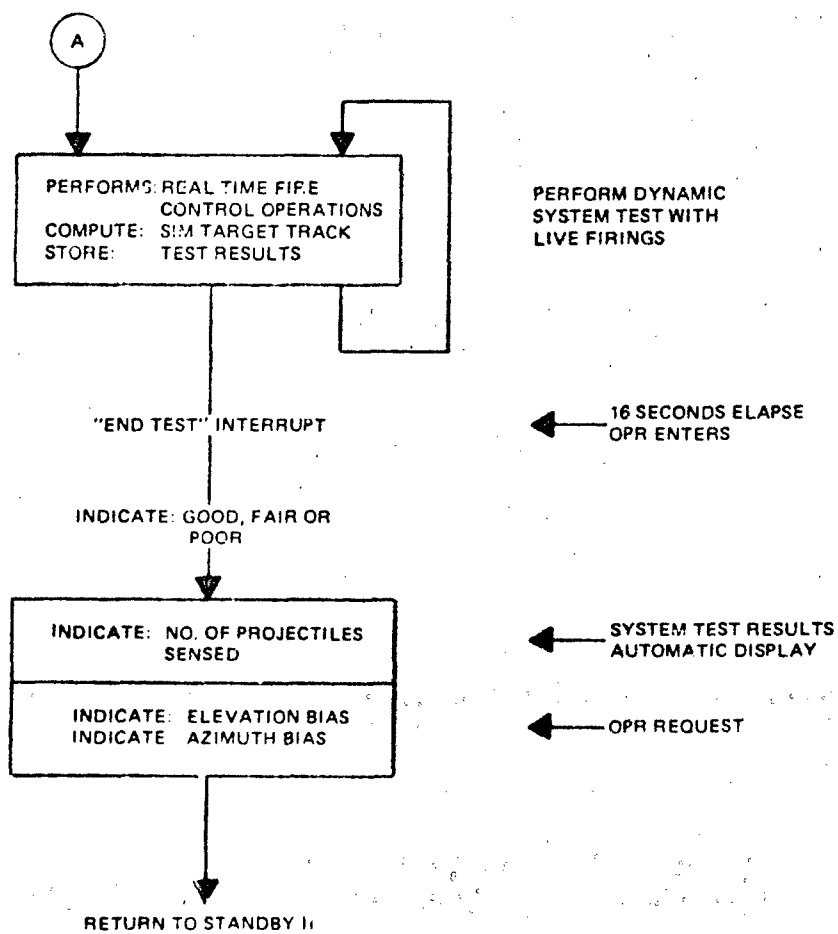
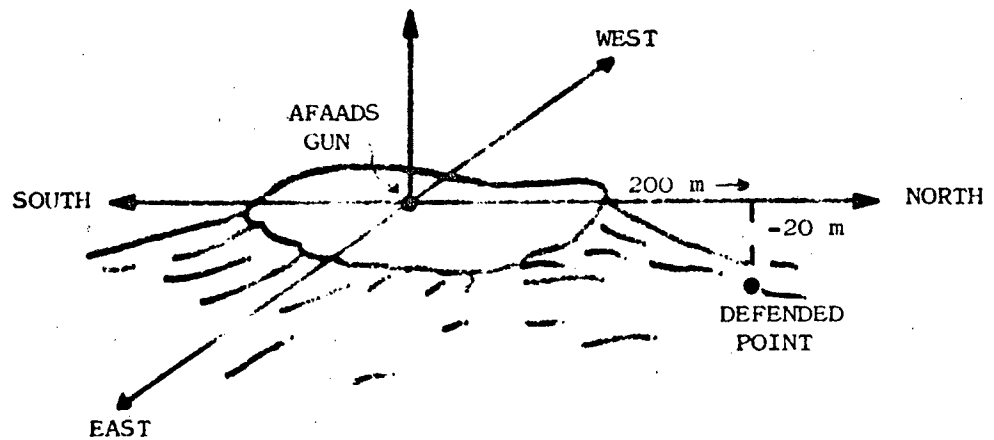


Figure 13-2. Medium Level Test Logic Flow Diagram (Sheet 2 of 2).



AFAADS GUN

Location: On Hill

Vehicle Orientation: Face North

Site: Level

DEFENDED POINT (SIMULATED)

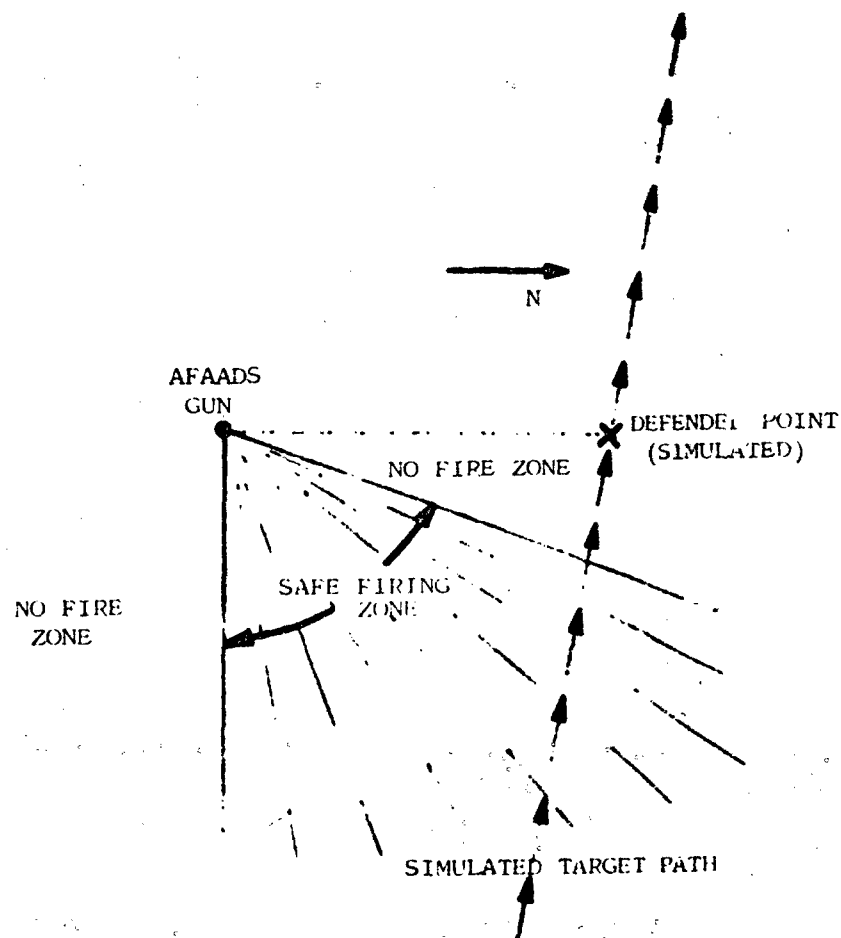
Location: In Valley

Distance: 200 m from AFAADS

Elevation: 20 m below AFAADS

Bearing: 0 Deg. (T) from AFAADS

Figure 13-3. System Test Situation



#### TARGET TRAJECTORY

DIRECTION: From right (East)  
 ALTITUDE: 200 m constant  
 SPEED: 300 m/sec constant  
 PT OF CLOSEST APPROACH: 200 m  
 (over Defended Point)

#### TEST FIRING

LOCATION: In safe firing zone  
 DURATION: 16 sec  
 INITIAL: Az  $87.8^{\circ}$ , El  $1.7^{\circ}$   
 FINAL: Az  $83.2^{\circ}$ , El  $5.4^{\circ}$

Figure 13-4. System Test Firing Area

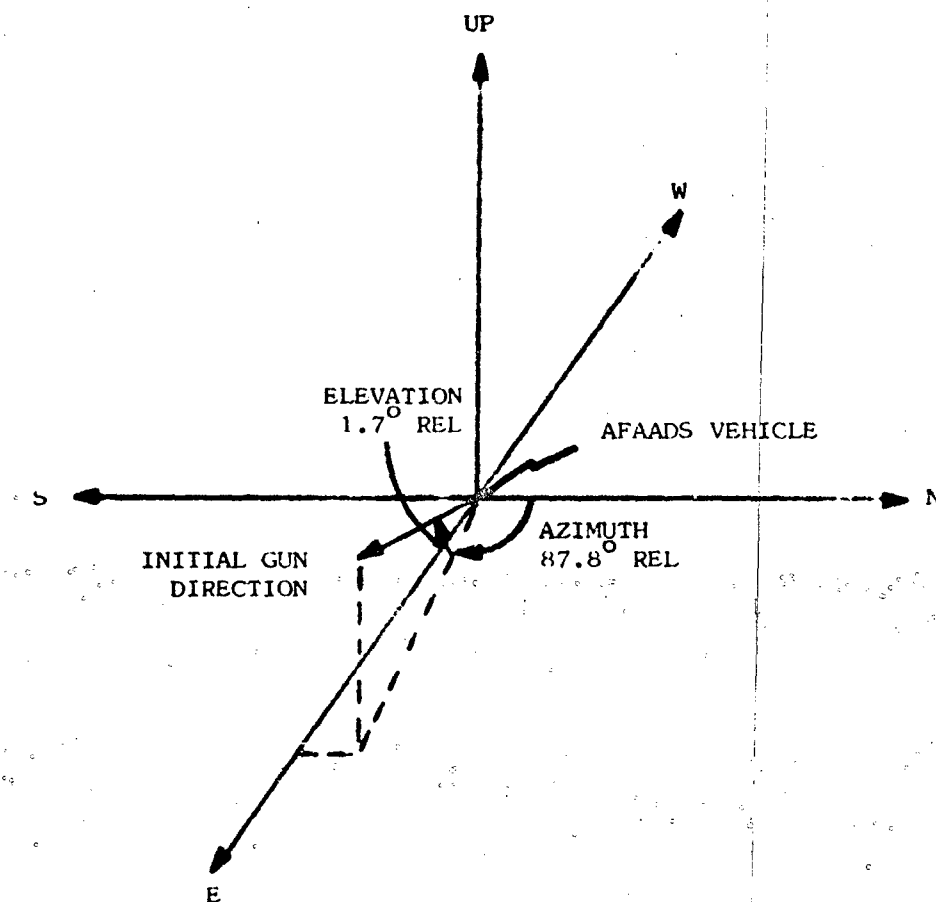


Figure 13-5. Coordinates for System Test Situation

A closer examination of these two figures is necessary to gain a fuller appreciation of the applicability of the dynamic system test.

Neglecting for a moment the safety features of the illustrated example, the simulated target trajectory of Figure 13-4 can provide a test of both simple and extreme conditions. High angular accelerations occur during the two second area about the point of closest approach. (Explained in more detail in the Typical Fire Mission discussion of Subsection 5.2, which uses the same geometry.) If an operator cares to stress severely the system for a short time (and safety considerations permit it), he may set the initial gun/sensor azimuth and elevation so the simulated target is relatively close to the gun. Thus the high angular velocities and accelerations (around 1600 mils/sec and 1600 mils/sec<sup>2</sup>) develop to strain the AFAADS gun and sensor servos.

In the particular case chosen, however, the target is relatively distant, allowing the maximum length of projectile-seeing time. The particular test case described allows almost eight seconds of miss sensing. (Given an initial elevation of about 1.7 degrees and azimuth of 87.8 degrees, the "target" flies from a range of 6600m to 1800m in a sixteen second test.) Almost half of this particular test is spent in projectile sensing. However, the azimuth rate of the gun is never greater than a degree per second.

The test is designed to run for sixteen seconds, which may or may not be long enough to test fully the AFAADS gun. In the "Safe Firing Area" of Figure 13-4 a sixteen second test will have the azimuth change by about four degrees, with the elevation angle also changed by about the same amount. But, the total "Safe Firing Area" is about seventy degrees wide. This would allow a total test time of almost 22 seconds. The gun and sensor rates would become quite high by the time the end of this region is reached.

We shall now return to the general discussion of the dynamic system test procedures and logic flow. After the Operator presses INIT SYSTEM TEST (Initiate System Test) on the Operator's Panel, the CONFID TEST (Confidence Test) indicator goes out. The operator may then enter

"where" the computer should place the simulated target at the beginning of the test run and "how" the target should fly. The test procedure is such that he first enters TARGET FROM LEFT or TARGET FROM RIGHT to indicate the direction which the target will take in passing the AFAADS gun. Then he enters the initial angular position of the gun and sensor by using ENTER E and ENTER A pushbuttons and the desired numeric enters on the keyboard. The initialization for system test is now complete.

The next phase of the test is an actual use of the complete AFAADS system. Once BEGIN SYSTEM (dynamic) TEST is depressed, a half-second of calculation takes place. The results of the computations are simulated target position and velocity at the beginning of the test, at the azimuth and elevation values just entered. Then the sensors and gun begin to move, following the "canned" target the computer generates. For sixteen seconds the sensors and gun slew, following this "target". The operator waits for FIRE BEGIN to be indicated, then fires the weapon. He will probably do his own monitoring of the test ammunition if he uses tracer ammunition. Otherwise, he will use the computer results.

Should he choose to interrupt the test after less than sixteen seconds, he may press END TEST which stops the testing. The results would then be displayed. Otherwise, he may complete the entire 16 seconds of testing and wait for the display of the results.

#### 13.2.4 System Dynamic Test Results

The indication of system dynamic test results can be a four part operation. An automatic display is presented immediately following the end of the test. This will show whether the mean projectile miss distance was "good", "fair", or "poor", by lighting the corresponding indicator. A "good" result is defined when the root-mean-square of the mean transverse and elevation angular projectile miss distances is less than one standard deviation of the dispersion (See Section 11.3, Block XVI, Item 4.3). Thus a "good" test shows that the average observed projectile passed within 3 mils of the simulated target. A "fair" test has the average projectile passing the target within the ring of 3 mils

and 6 mils. A "poor" test has a miss of greater than 6 mils.

Several distribution patterns are shown in Figure 13-6 ranging from "good" through "poor". Example 7 in the figure illustrates the situation where an inadequate number of projectiles are sensed to obtain meaningful results. In this case "poor" test results would also be displayed.

To determine the cause of "fair" or "poor" test results, the Operator is provided with three display buttons. The number of projectiles sensed can be observed by first pressing NO OF PROJECT (Number of Projectiles) and reading the value in the LED display on the keyboard. The mean azimuth and elevation bias errors can be displayed through use of the ELEV BIAS (Elevation Bias) and AZM BIAS (Azimuth Bias) pushbuttons and the LED display. These values which are signed, designate the direction of skew or bias of the average projectile observed, and hence show the angular error of the average projectile during the test.

Having finished reviewing the test results, the operator will normally switch the system into Standby II by pressing STBY II.

### 13.3 Growth in Self-Test Capabilities

Better tests may be added at a later date. The present tests are minimal, but more comprehensive tests may place a great strain on the system. For instance, one test requiring the calculation of the standard deviation of the shots sensed, would require a large amount of computer storage, but would certainly be desirable. Addition of comprehensive testing might well place a greater load on the AFAADS than an actual engagement - an ironic and in some ways an undesirable result.

More study with a real system could show that the testing ought to be extensive. Cost-effectiveness of the concept might prove that a larger computer with excellent test capabilities might well insure that the entire AFAADS gun system could almost maintain itself, lucrative to the man in the field in both kill capabilities and "down time" reductions. Presently, one can only conjecture. The present test

### TEST RESULTS

1. Results: "GOOD"

Azimuth Error:  $\leq 3$  mils

Elevation Error:  $\leq 3$  mils

No. of Shots Seen:  $> 15$

2. Results: "FAIR"

Azimuth Error:  $\leq 3$  mils

Elevation Error:  $\geq 3$  mils,  
 $\leq 6$  mils

No. of Shots Seen:  $> 15$

3. Results: "FAIR"

Azimuth Error:  $\geq 3$  mils,  
 $\leq 6$  mils

Elevation Error:  $\leq 3$  mils

No. of Shots Seen:  $> 15$

4. Results: "POOR"

Azimuth Error:  $\leq 3$  mils

Elevation Error:  $\geq 6$  mils

No. of Shots Seen:  $> 15$

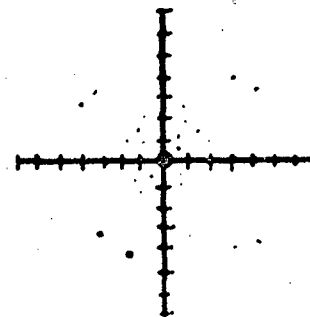
### SHOT PATTERN

1. "GOOD"

$\Delta A$ : 0.1

$\Delta E$ : 0.0

N: 20

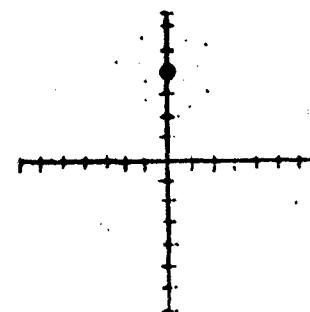


2. "FAIR"

$\Delta A$ : 0.1

$\Delta E$ : 4.0

N: 17

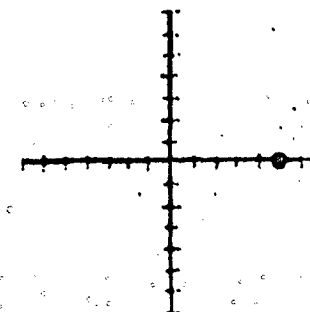


3. "FAIR"

$\Delta A$ : 5.1

$\Delta E$ : -0.2

N: 23



4. "POOR"

$\Delta A$ : 1.5

$\Delta E$ : -6.2

N: 31

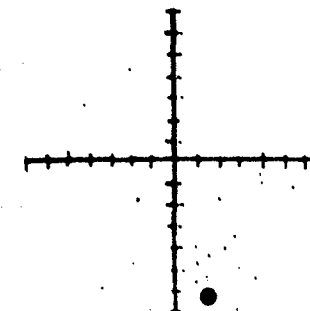


Figure 13-6. System Test Results (Sheet 1 of 2)

### TEST RESULTS

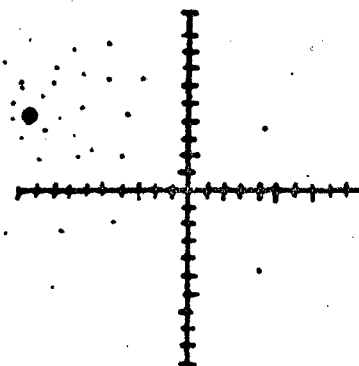
### SHOT PATTERN

5. Results: "POOR"

Azimuth Error:  $\geq 6$  mils  
Elevation Error:  $\geq 3$  mils,  
 $\leq 6$  mils  
No. of Shots Seen:  $> 15$

5. "POOR"

$\Delta A$ : -7.2  
 $\Delta E$ : 3.3  
N: 28

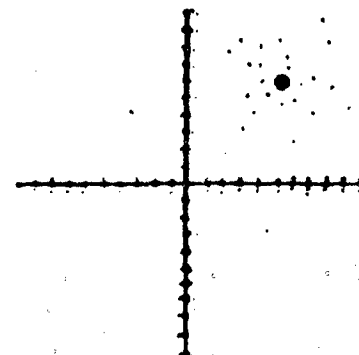


6. Results: "POOR"

Azimuth Error:  $\geq 3$  mils,  
 $\leq 6$  mils  
Elevation Error:  $\geq 3$  mils,  
 $\leq 6$  mils  
Total Error:  $\geq 6$  mils  
No. of Shots Seen:  $> 15$

6. "POOR"

$\Delta A$ : 4.5  
 $\Delta E$ : 4.6  
Total: 6.4  
N: 27



7. Results: "POOR"

Azimuth Error: ?  
Elevation Error: ?  
No. of Shots Seen:  $\leq 15$   
("POOR" because too few projectiles seen  
for valid test results)

7. "POOR"

$\Delta A$ : -1.0  
 $\Delta E$ : -0.5  
N: 8

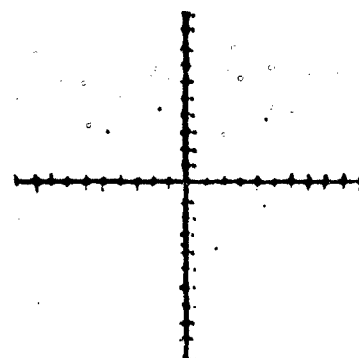


Figure 13-6. System Test Results (Sheet 2 of 2)

procedure provides simply-calculated but highly important data to the Operator - giving the Gunner a "good idea" as to what is right or wrong with his gun.

## APPENDIX A

### EXCERPTS FROM STATEMENT OF WORK

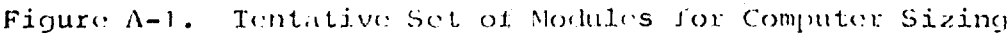
The conceptual design for the digital computer for AFAADS was performed in fulfillment of Task 1D and 1E of the Statement of Work. These task statements are repeated below.

#### SUBTASK 1.D Software Requirements Definition

The software package for the miss distance correction system as well as the basic fire control data processing functions will be defined to a level which will allow detailed evaluation of software requirements and estimation of the required computer characteristics, including architecture, size, and approximate cost. The required data rate for successful system operation will be determined as a part of this effort. Emphasis will be placed on modular software design for maximum system flexibility and growth potential.

A preliminary designation of the principal data processing functions is shown in Figure A-1. Since not all of these functions may be finally employed in a design development, the computer program structure should be modular to allow each function to be performed by a separate routine. The object is to provide a baseline system configuration to which modules providing additional or different capability may be added when desired.

This modular concept is considered particularly valuable in the case of the prediction mode options, where, for example, it would be highly desirable to field test an experimental system with a variety of prediction modes simply by employing different program modules. Likewise, in the case of the ballistic module, the same computer might be used with several weapon types by changing the ballistic program module. This approach would also facilitate system algorithmic change as new enemy tactics are encountered. The projectile miss distance correction algorithms would likewise be designed as an add-on module, which could be changed as different or improved miss distance sensors became available.



The data flow for the projectile miss data processing and correction algorithms is detailed in Figure A-2 for a "brute force" solution, and in Figure A-3 for a solution employing a reference computed trajectory. A preliminary analysis of both of these concepts was provided in the previous final report.

Software definition will include a definition of all computer program functions, overall organization and structure of the software, formulas, computational procedures, and flow charts of computations, and estimates of program and data storage requirements.

#### SUBTASK 1.E Computer Requirements Definition

The methodology of computer requirements definition is illustrated in Figure A-4. Computer requirements will be developed as follows:

- a. Software requirements along with system operational requirements and equipment physical requirements provide the basis for analyzing and defining computer implementation. Since the AFAADS gun system has not yet been specified in detail, the computer requirements in many respects will be tentative and may cover a range of required functions and system interfaces. Typical requirements must be defined as a part of this task. Software requirements will be defined in Subtask 1.D in the form required to evaluate the effectiveness of various computer architectures and operating characteristics. System operational requirements include requirements for interfacing with the weapon and sensor systems and with operator display and control devices. Other possible interfaces such as those with target identification equipment and data transmission equipment for coordination and control of forward area weapons are also to be considered. A host of additional requirements relating to the modes of employment and operation of the system must also be defined. In addition, physical requirements must be postulated consistent with the intended application of the computer. These would include environmental, EMI/TEMPEST,

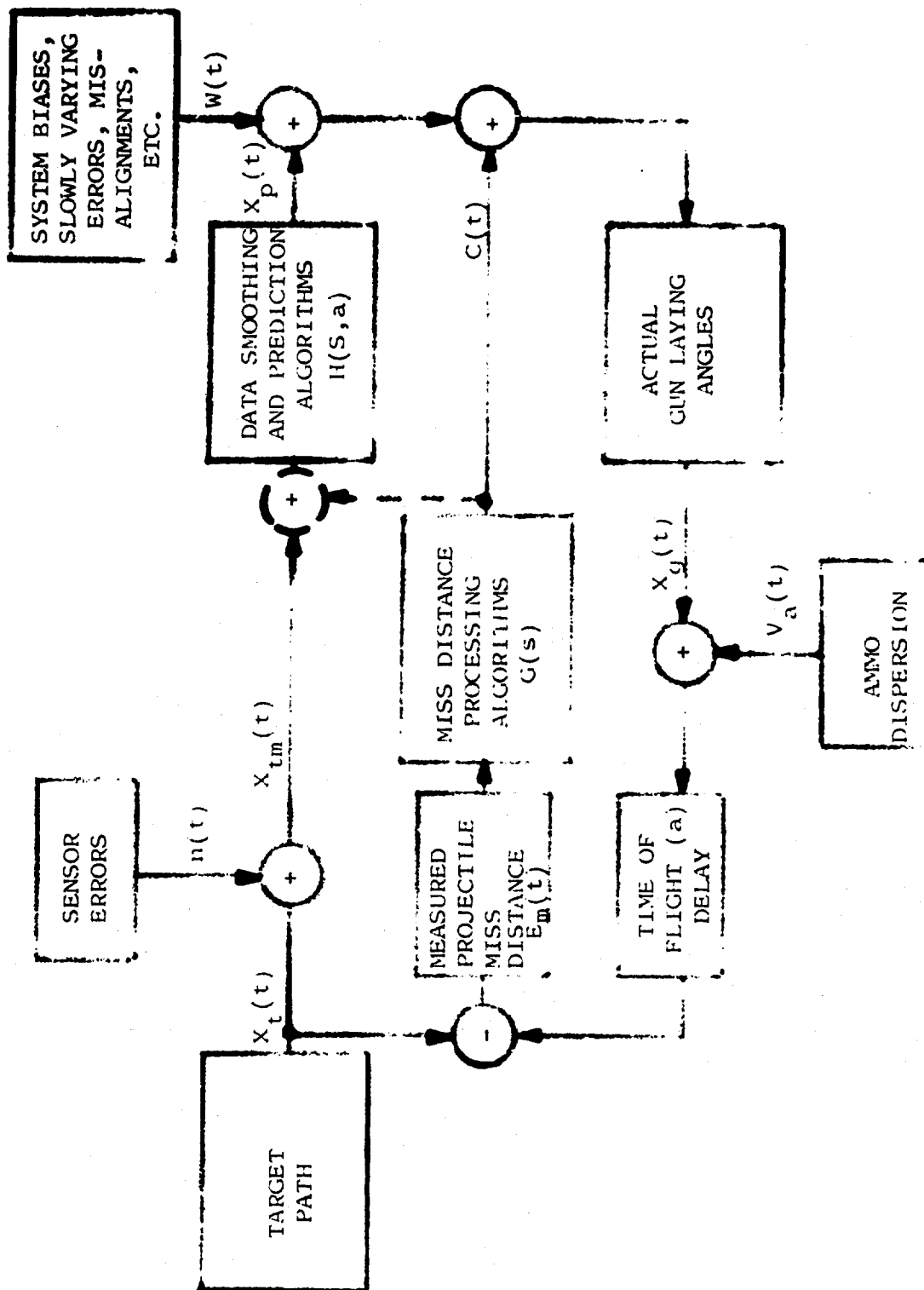


Figure A-2. Flow Diagram for System Correction Based on Measurements of Projectile Miss Distances



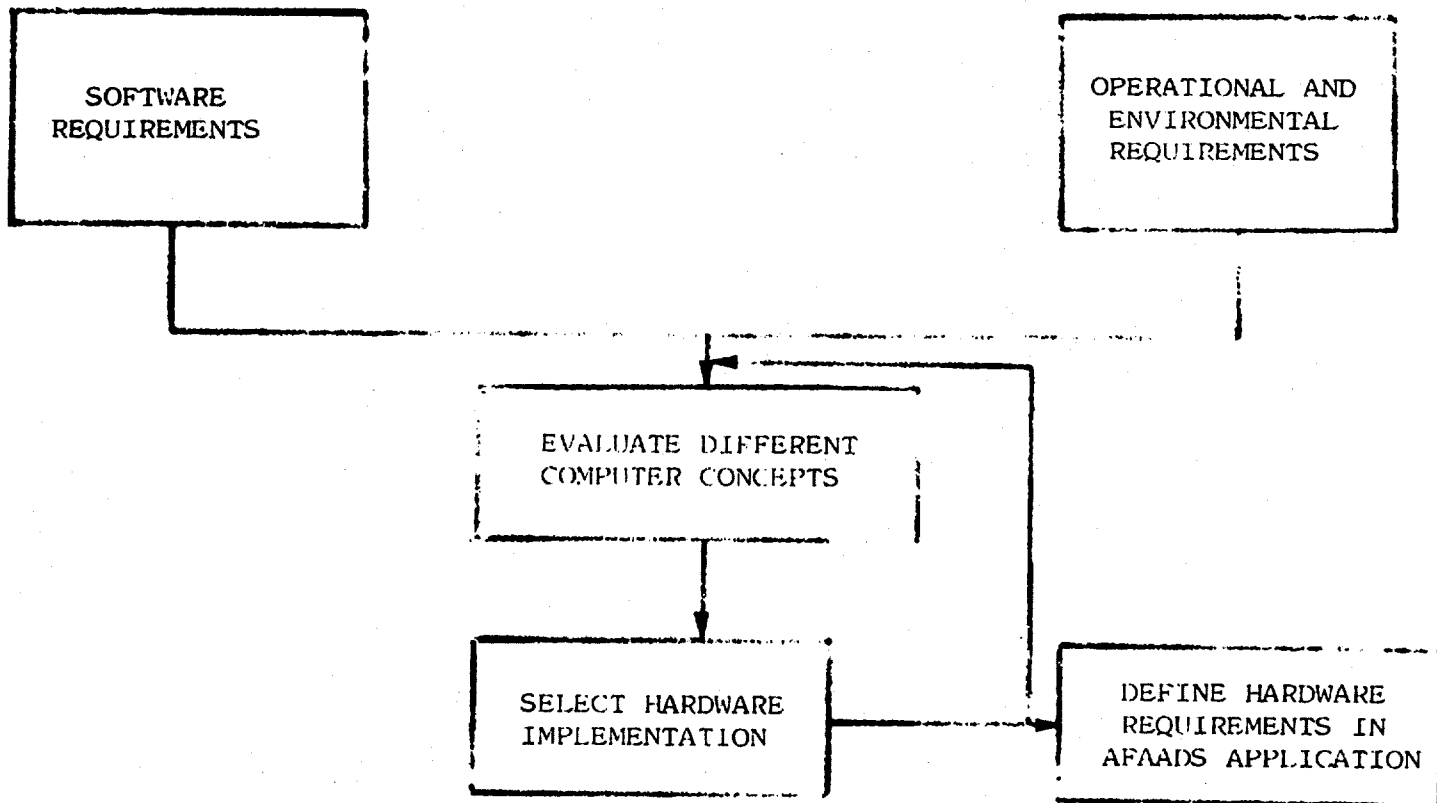


Figure A-4. Computer Requirements Definition

reliability, maintainability and logistic support requirements as well as size, weight and power requirements. Operational and physical requirements will be delineated as the first step in defining computer hardware requirements.

- b. Various computer implementation concepts will be evaluated against the system requirements to determine the most suitable general form of computer implementation. Candidate concepts will be based on state-of-the-art digital computer technology. Because of the nature of the fire control problem it may be appropriate to consider a special-purpose computer implementation to provide a highly efficient solution. However, the availability and flexibility of general purpose computers may make this approach more desirable. The study will consider the relative advantages of different computer techniques and organizations. These considerations include conventional arithmetic versus special purpose logic, a single central processing unit versus distributed functions, conventional versus microprogram control, input/output system organization and operation, and memory type and organization. Evaluation of these basic concepts will provide the basis for defining one or more approaches to computer implementation using available technology.
- c. Hardware implementation of one or more computer concepts will be defined based on known available computer components. Since the fire control problem represents a small computer requirement, the study is expected to benefit from the current concentration in the computer industry on mini- and micromini-computers. The study will give preference to existing or developmental computers as opposed to entirely new designs. Hardware implementations will be further evaluated in accordance with system requirements and one or more implementations selected.

- d. Based on the selected implementation(s), a definition of hardware and its application to the AFAADS fire control problem will be produced. The hardware definition will include description of the computer hardware required for computation of the problem and for interfacing with other components of the system. Computer requirements definition produced in this subtask will permit evaluation of the operational effectiveness and cost effectiveness of digital fire control computer implementation for AFAADS and will provide the basis for defining an experimental prototype computer if desired.

## APPENDIX B

### AFAADS FINITE LENGTH TRACKING FILTERS

#### B.1 Introduction

This appendix presents the target tracking algorithms that were chosen for use in the AFAADS computer sizing task. The chosen algorithms are of finite or fixed length, rather than of infinite length based on recursive formula algorithms.\* In addition this appendix also presents the error variance reduction ratios between the filter outputs of target position, velocity, etc. relative to raw target position input data.

The formulas are quite general. They provide polynomial smoothing of the input data to any desired degree (linear or velocity smoothing, quadratic or acceleration smoothing, etc.). Also the filter output provides the best estimate, in a least squares sense, of the target's position, velocity, etc. relative to any desired time interval. Thus the best estimate of position can be chosen relative to (a) the midpoint of the smoothing interval, (b) the last or current position, (c) any future or predicted time, or (d) any other time desired. The error variance reduction ratios can similarly be computed for any time point.

Following the presentation of the general formulas, they are first applied to the specific example of linear or velocity smoothing. These are then further simplified to give the best estimate of last measured position of the target. The last measured position rather than the midpoint in the smoothing interval was chosen since the AFAADS gun system will be tracking a target flying an unpredictable and maneuvering course. Last measured position filtering weighs the later points more heavily than the early ones and hence should more truly reflect the aircraft's true motion at the end of the sampling interval. It is only the last

\*Comparison with adoptive Kalman filters and other tracking techniques should be included in the next phase of AFAADS.

measured position and future or predicted positions that are required by the other real time fire control algorithms. (Sec. 11). This choice of using the last measured position instead of the midpoint was made at the expense of slightly increasing the error reduction ratio. (Note that if the target were flying on a defined trajectory, such as an orbiting space vehicle, use of the midpoint would provide more accurate data on which the specific trajectory could be defined.)

Finally the specific tracking formulas are presented for linear (velocity) smoothing over a one second time interval using data every 0.1 sec. This is the specific tracking situation used in the AFAADS computer concepts analysis, specifically Block II of Sec. 11.3.

In the implementation of AFAADS, it is proposed that the sensor (FLIR/laser or tracking radar) be mounted on a rate-gyro stabilized head. Such a sensor head will provide both position data (range, azimuth, and elevation) and angular rate data (azimuth and elevation rates).

Assuming that either doppler data or accurate difference range data are available, the inputs to the tracking filters will then be three dimensional position data and three dimensional rate data. Under these conditions the target tracking algorithms should be used twice:

- (1) To reduce the raw position data to smoothed position data.
- (2) To reduce the raw rate data to both smoothed rate data and smoothed acceleration data.

This is the assumed situation in the analysis of this volume of AFAADS.

## B.2 Reference Material

The tracking algorithms proposed for AFAADS are those developed by Norman Morrison in "Introduction to Sequential Smoothing and Prediction, Chap. 7 (Ref. 4). The nomenclature of Morrison has been modified to conform to that used in this AFAADS report. However, specific references are made to Morrison's analysis through the use of his equation numbers on the right margin.

### B.3 Processing Situation

Assume that the AFAADS sensor head (FLIR/laser or radar) provides range, azimuth, and elevation position data (no rate data) to the computer once every  $\Delta t$  seconds. The last data received are at clock time  $t(n)$ , the previous data at  $t(n-1)$ . These data are converted to Cartesian Coordinates  $x$ ,  $y$ , and  $z$  every sampling period and become the inputs to the target tracking filter. Since the filter will process the data in each coordinate in exactly the same manner, only the  $x$ -component will be treated below.

The tracking algorithm makes the following additional assumptions:

- (1) Filter smooths with a polynomial of degree  $m$  ( $m$  equals 1 for velocity smoothing, 2 for acceleration smoothing, etc.)
- (2) Filter is of finite length covering data over the last  $L \Delta t$  seconds.
- (3)  $L+1$  data points are processed each time.
- (4) Last data received and processed is  $x(n)$  at time  $t(n)$ . Oldest data point processed in filter is  $x(n-L)$  received at time  $t(n-L)$ .
- (5) Filter output position  $\underline{x}(n+p)$  gives the best estimate of position  $p\Delta t$  seconds after the last measured position. Note  $p$  can be positive (prediction), zero (current position) or negative (back in time).
- (6) Filter output velocity is  $\dot{\underline{x}}(n+p)$  or  $D\underline{x}(n+p)$ . Higher order derivatives are also used.

#### B.3.1 General Filter Equations

Matrix notation (capital letters) is used extensively in order to simplify the presentation. Let the total observation vector or the vector describing all the raw data points used by the tracking filter be designated by:

$$X(n) = \begin{pmatrix} x(n) \\ x(n-1) \\ \dots \\ x(n-L) \end{pmatrix} \quad (7.3.25)$$

Similarly the best estimate vector or the vector describing the filter output is

$$\underline{X}(n+p) = \begin{pmatrix} \underline{x}(n+p) \\ D\underline{x}(n+p) \\ \dots \\ D^m \underline{x}(n+p) \end{pmatrix} \quad (7.4.7)$$

where  $m$  is the order of the polynomial used by the filter and also equals the highest order derivative of  $\underline{x}$  that can be computed.

Morrison's analysis shows that:

$$\underline{X}(n+p) = W(p, \Delta t)X(n) \quad (7.4.11)$$

where  $W(p, \Delta t)$  is the unscaled weighting function matrix.

The remainder of the general analytic results describe this weighting matrix in terms of other matrices and thence in terms of defined quantities.

$$W(p, \Delta t) = D(\Delta t)W(p) \quad (7.4.11 \text{ and } 7.4.12)$$

where:  $D(\Delta t)$  is the diagonal scaling matrix with elements:

$$\left[ D(\Delta t) \right]_{ij} = \left( \frac{j}{\Delta t^j} \right) \delta_{ij} \quad (4.2.19)$$

where:

$$0 \leq i, j \leq m$$

$\delta_{ij}$  is the Kronecker delta

$m$  is the order of the polynomial used.

$W(p)$  is the scaled weighting function matrix.

The scaled weighting function matrix is the product of five matrices:

$$W(p) = P(p) \text{ SGC B} \quad (7.4.16)$$

where: (1)  $P(p)$  is the transition matrix (time prediction matrix) with elements:

$$[P(p)]_{ij} = \binom{j}{i} p^{j-i} \quad (4.2.15)$$

where:

$$0 \leq i, j \leq m$$

$p$  is the number of prediction time intervals  $\Delta t$  relative to current time  $t(n)$ . It need not be an integer, and can be positive, zero, or negative.

(2)  $S$  is the link matrix or associate Stirling matrix of the first kind. The elements of this matrix must be computed by recursive formulas (See Morrison P. 20-Ref. 4). The first few terms are listed in the following table:

i	j				
	0	1	2	3	4
0	1	0	0	0	0
1	0	1	1	2	6
2	0	0	1	3	11
3	0	0	0	1	5
4	0	0	0	0	1

(Table 4.1)

- (3)  $G$  is a matrix dealing with the length of the filter and has terms:

$$[G]_{ij} = (-1)^j \binom{j}{i} \binom{j+i}{i} \frac{1}{L^{(i)}} \quad (7.4.14)$$

where:  $0 \leq i, j \leq m$

$$L^{(i)} = L(L-1)(L-2) \dots (L-i+1) \quad (2.4.1)$$

This function  $L^{(i)}$  is called the backward factorial function. This function also occurs in other matrices below.

- (4)  $C$  is the matrix of normalizing factors for the discrete Legendre polynomials. It is diagonal with terms:

$$[C]_{ij} = \left( \frac{1}{c_j^2} \right) \delta_{ij} \quad (7.4.5)$$

where:  $0 \leq i, j \leq m$

$$c_j^2 = c^2(j, L) = \frac{(L+j+1)(j+1)}{(2j+1)L^{(j)}} \quad (7.3.4)$$

This function  $c_j$  is the normalizing factor for the  $j$ 'th order polynomial.

- (5)  $B$  is the only non-square matrix. It has terms

$$[B]_{ij} = p_i(x) \Big|_{x=L-j} \quad (7.4.6)$$

where:  $0 \leq i \leq m$

$0 \leq j \leq L$

$$P_i(r) = p(r; j, L) = \sum_{v=0}^j (-1)^v \binom{j}{v} \binom{j+v}{v} \left( \frac{r^{(v)}}{L^{(v)}} \right) \quad (3.2.20)$$

The function  $p_j(r)$  is the discrete Legendre polynomial.

### B.3.2 General Variance Reduction Formula

Morrison's analysis also considers error variance reduction ratios between the filter outputs and inputs. These ratios are based upon the following assumptions with regard to the errors in the input data  $x(n)$ :

1. Errors have zero mean, i.e. no bias errors.
2. Errors on each data point are uncorrelated with those on any other data point.
3. All errors have the same standard deviation  $\sigma(x)$

This means that the covariance vector of the input data errors can be expressed by

$$R(n) = \sigma^2(x) I \quad (7.5.5)$$

where  $I$  is the identity matrix.

Under these conditions the best estimate vector on the output of the filter will have an error variance matrix  $S(n+p)$  whose terms are given by:

$$[S(n+p)]_{ij} = \frac{\sigma^2(x)}{\Delta t^{i+j}} \sum_{k=0}^m \frac{d^i \phi_k(r)}{dr^i} \frac{d^j \phi_k(r)}{dr^j} \Big|_{r=L+p} \quad (7.5.28)$$

where:  $0 \leq i, j \leq m$

$$\phi_k = p_k(r)/c_k \quad (7.3.6)$$

and both  $p_k(r)$  and  $c_k$  have already been defined.

#### B.4 Example of Linear or Velocity Smoothing (m equals 1)

The above general equations are now applied to the example of linear or velocity smoothing. Hence m equals 1. Only the first two discrete Legendre polynomials and their normalization factors are required; namely,

$$p_0(r) = 1 \quad p_1(r) = 1 - (2r/L) \quad (3.2.21)$$

$$c_0^2 = L+1 \quad c_1^2 = (L+2)(L+1)/3L \quad (3.2.31)$$

For linear smoothing the output of the tracking filter gives the best estimate of the position of the target at a prediction interval p, which occurs at a time t(n+p)

$$\begin{aligned} \underline{x}(n+p) = & \frac{1}{L+1} \left\{ \left[ 1 + \frac{3(L+2p)}{L+2} \right] x(n) + \left[ 1 + \frac{3(L+2p)(L-2)}{L(L+2)} \right] x(n-1) \right. \\ & + \left[ 1 + \frac{3(L+2p)(L-4)}{L(L+2)} \right] x(n-2) + \dots \\ & \left. + \left[ 1 - \frac{3(L+2p)}{L+2} \right] x(n-L) \right\} \end{aligned}$$

The velocity output of the filter is:

$$\begin{aligned} \dot{\underline{x}}(n+p) = D \underline{x}(n+p) = & \left\{ 6/[\Delta t L(L+1)(L+2)] \right\} \left\{ Lx(n) + (L-2)x(n-2) \right. \\ & \left. + (L-4)x(n-4) + \dots -Lx(n-L) \right\} \end{aligned}$$

Under these circumstances, the error variance reduction ratio matrix becomes:

$$\begin{aligned} S(n+p) &= \begin{pmatrix} \sigma^2(\underline{x}(n+p)) & \sigma^2(\underline{x}(n+p), \dot{\underline{x}}(n+p)) \\ \sigma^2(\underline{x}(n+p), \dot{\underline{x}}(n+p)) & \sigma^2(\dot{\underline{x}}(n+p)) \end{pmatrix} \\ &= \frac{2\sigma^2(\underline{x})}{(L+2)(L+1)(L)} \begin{pmatrix} L(2L+1) + 6p(L+p) & 3(L+2p)/\Delta t \\ 3(L+2p)/\Delta t & 6/(\Delta t)^2 \end{pmatrix} \end{aligned}$$

This matrix equation gives the error variance reduction ratio for position (upper left hand terms) and for velocity (lower right hand terms). The other two terms are cross correlation reduction ratios.

#### B.4.1 Linear Smoothing for Last Measured Position ( $p=0$ )

To consider the best estimate of the last measured position of the target, set  $p$  equal to zero in the above equations. This gives the smoothed position, smoothed velocity, variance reduction in position, and variance reduction for velocity as follows:

Output position:

$$\underline{x}(n) = 2[(2L+1)x(n) + (2L+1-3)x(n-1) + (2L+1-6)x(n-2) + \dots + (2L+1-3g)x(n-g) + \dots + (1-L)x(n-L)] / (L+2)(L+1)$$

Output velocity:

$$\dot{\underline{x}}(n) = 6[Lx(n) + (L-2)x(n-1) + (L-4)x(n-2) + \dots - Lx(n-L)] / \Delta t(L+2)(L+1)$$

Output position error reduction:

$$\sigma^2(\underline{x}(n)) = \left( \frac{2(2L+1)}{(L+2)(L+1)} \right) \sigma^2(x)$$

Output velocity error reduction:

$$\sigma^2(\dot{\underline{x}}(n)) = \left( \frac{12}{(\Delta t)^2 (L+2)(L+1)(L)} \right) \sigma^2(\dot{x})$$

#### B.4 One Second Filter

Finally we will consider the specific algorithm for a one second filter using data received every 0.1 second. For this case:

$$\Delta t = 0.1 \text{ second}$$

$$L = 10$$

$$L \Delta t = 1.0 \text{ second}$$

(This is the specific case used in the AFAADS computer concept analysis of this Volume of the AFAADS Final Report).

Output position:

$$\underline{x}(n) = [7 x(n) + 6 x(n-1) + 5 x(n-2) + 4 x(n-3) + 3 x(n-4) + 2 x(n-5) + x(n-6) + 0 x(n-7) - x(n-8) - 2x(n-9) - 3x(n-10)]/22$$

Output velocity:

$$\dot{\underline{x}}(n) = D \underline{x}(n) = [5 x(n) + 4 x(n-1) + 3 x(n-2) + 2 x(n-3) + x(n-4) + 0 x(n-5) - x(n-6) - 2 x(n-7) - 3 x(n-8) - 4 x(n-9) - 5 x(n-10)]/11$$

Output position error reduction:

$$\sigma(\underline{x}(n)) = \sqrt{0.318} \sigma(x) = 0.564 \sigma(x)$$

Output velocity error reduction:

$$\sigma(\dot{\underline{x}}(n)) = \sqrt{0.910} \sigma(\dot{x}) = 0.955 \sigma(\dot{x})$$

If midpoint smoothing had been chosen, the position error reduction ratio would be 0.302 and the velocity error reduction ratio the same, or 0.955.

## APPENDIX C

### BALLISTIC CORRECTION FOR DOWN WIND

One of the ballistic corrections to be applied in the AFAADS fire control system is the effect of a down (or up) wind on the projectile. Such a wind will cause the projectile to pass beyond (or in front of) the target.

The geometric situation is shown in Figure C-1. In the absence of wind, the AFAADS gun will be fired to the predicted target range  $D(p)$  at an elevation angle  $E(p)$ . Because of a down wind the projectile will travel farther down range than predicted and will miss the target by a distance  $\Delta D(w)$  as measured at the target's altitude. This is equivalent to a small angular elevation error  $\phi(w)$ . If the gun elevation were increased by  $\phi(w)$ , the wind would curve the projectile's path so as to intercept the target.

Analytically by the law of sines

$$\frac{\sin \phi(w)}{\Delta D(w)} = \frac{\sin \eta}{D(p)}$$

From the figure

$$\eta = E(p) - \phi(w)$$

Substituting and assuming that the wind correction factor is small, i. e., a first order correction

$$\phi(w) = \Delta D(w) \sin E(p) / D(p)$$

For the AFAADS computer sizing task it is assumed that:

- a) The form of the down-wind and cross-wind correction factors are the same (Reference 1, Volume 1, paragraph 5.5.4 on page 5-53).
- b) The ratio of the two correction factors is that given in Reference 1, Volume 1, paragraph 5.5.4; namely, the down-wind linear miss is 1.5 times the cross-wind linear miss; and
- c) The explicit cross-wind correction factor, developed in Appendix G, can be used directly for the down-wind correction.

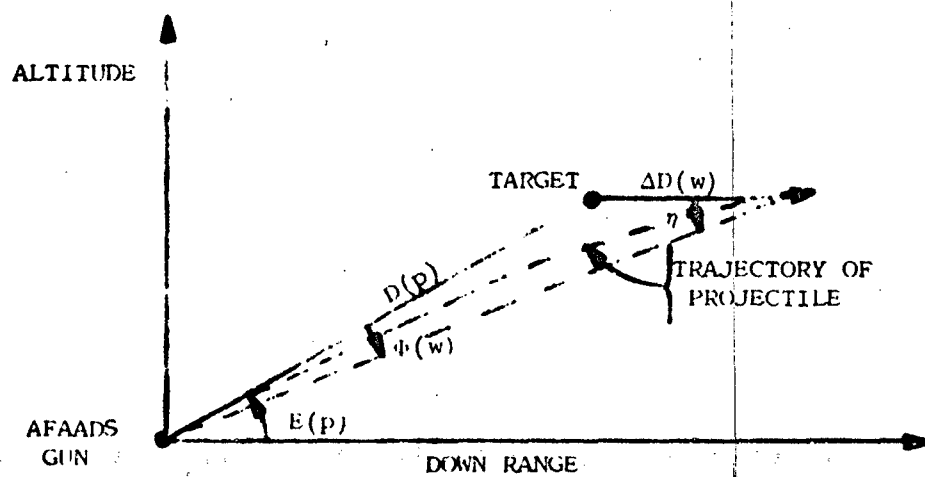


Figure C-1. Effect of Down Wind on the Projectile's Flight

Combining these assumption

$$\Delta D(w) = 1.5 \Delta L D(p)$$

where:  $\Delta L = KW(d) t(p) \sec E(p)[1-k t(p) \sin E(p)]$

= cross wind angular correction factor from Volume I  
except the down wind  $W(d)$  is used rather than the  
cross wind  $W(c)$ .

The two constants have the approximate values from Appendix G, Figure G-4

$$K = 0.115 \text{ milliradian/m.}$$

$$k = 0.0293/\text{sec.}$$

Thus the super-elevation angle ballistic correction factor to be  
used in the AFAADS computer sizing task is:

$$\phi(w) = 1.5 K W(d) t(p) \tan E(p)[1-kt(p) \sin E(p)]$$

In using this equation, down wind is defined as the wind velocity from the  
gun toward the target. A positive down wind means the gun's elevation angle  
must be increased  $\phi(w)$  to compensate for the wind's effect. (Similarly an  
up wind will cause a gun depression correction.)

## APPENDIX D

### VALIDITY CRITERIA FOR FLIR/LASER PROJECTILE MISS DISTANCE DATA

#### D.1 Introduction

In the assumed AFAADS gun fire control system using a FLIR/laser target tracking head, not all of the projectile miss distance data readings should be employed in the gun bias error correction algorithms. The limitations result from the technical characteristics of the assumed FLIR/laser sensor system. These limitations and characteristics are first discussed followed by a presentation of three data validity criteria. Satisfaction of these criteria means the measured data can be used for gun bias error corrections. (Additional discussion on this subject can be found in Volume I, Section 3).

#### D.2 Limitations of FLIR/Laser Sensor System

For the projectile miss distance measurement problem, the assumed FLIR/laser sensor system differs from a radar sensor system in two important aspects. These differences result in the above-mentioned validity criteria.

1. Viewing Rate: The assumed FLIR/laser sensor system for AFAADS will obtain about 30 looks at the target each second. This compares with the several hundred looks per second for a radar. Thus a projectile angular miss distance measurement with a radar can be obtained at much closer to the equal range point of target and projectile than is possible with a FLIR/laser sensor. To apply the equal range criteria used with radar to a FLIR/laser sensor system would mean that most potential miss distance measurements would be missed.
2. Data Characteristics: A radar obtains both range and angle data on a target (or projectile) simultaneously and from the same received signal. The assumed FLIR/laser sensor differs significantly. The laser range data is based upon a reflected signal. The FLIR signal is normally from radiation generated by the target itself - temperature differences, augmented in the case of projectiles by tracers. Or, the projectile sensing may rely on laser echo signals. Thus the signals processed by the two sensors are usually independent of one another.

Coupling can be obtained through the use of high pulse rate CO<sub>2</sub> lasers, such that the FLIR signals are essentially range-gated. Such a system is visualized in the following detection criteria. If such a system does not prove to be practical, a two dimensional bias correction algorithm could be used (see Volume I, Section 3). The computational requirements of the latter are less than for three dimensional and good azimuth and elevation bias corrections are obtained; but not velocity bias.

### D.3 Data Validity Criteria

Three proposed projectile miss distance data validity criteria are developed.

D.3.1 First Criteria: Consider the situation in Figure D-1. The target is assumed to be crossing in front of the AFAADS gun. A projectile is fired on a perfect course to the target, i.e., no biases exist. Due to the finite scan rate of the FLIR, assume that angular projectile miss distance measurements are made at  $t_1$  before the projectile reaches the equal range point, and again at  $t_2$ , after the projectile passes the equal range point. If these data were used for bias error corrections, the first measurement would indicate a bias miss to the right and the second a miss to the left; when in actual fact no bias exists.

Let us see how this situation can be used to develop the first measurement validity criterion. Between times  $t_1$  and  $t_2$ , the target moves through an angle  $\Delta\theta$  as measured by the AFAADS sensor tracking servo. If this angle is less than the error  $\Delta\theta_0$  involved in measuring the projectile angular miss distance, the measured value will give a true or valid indication of the bias correction that must be applied to the AFAADS gun. These angles can be expressed in terms of angular rates since the angle  $\Delta\theta$  is executed in the time  $\Delta t$  between  $t_1$  and  $t_2$ . Thus the first validity criterion is

$$\omega \leq \omega_0$$

where:

$\omega$  = Target angular crossing rate

$\omega_0 = \theta_0 / (1/2) \Delta t$  = Projectile angular miss distance error divided by half the scan period of the FLIR (the factor of one-half results from the third criterion).

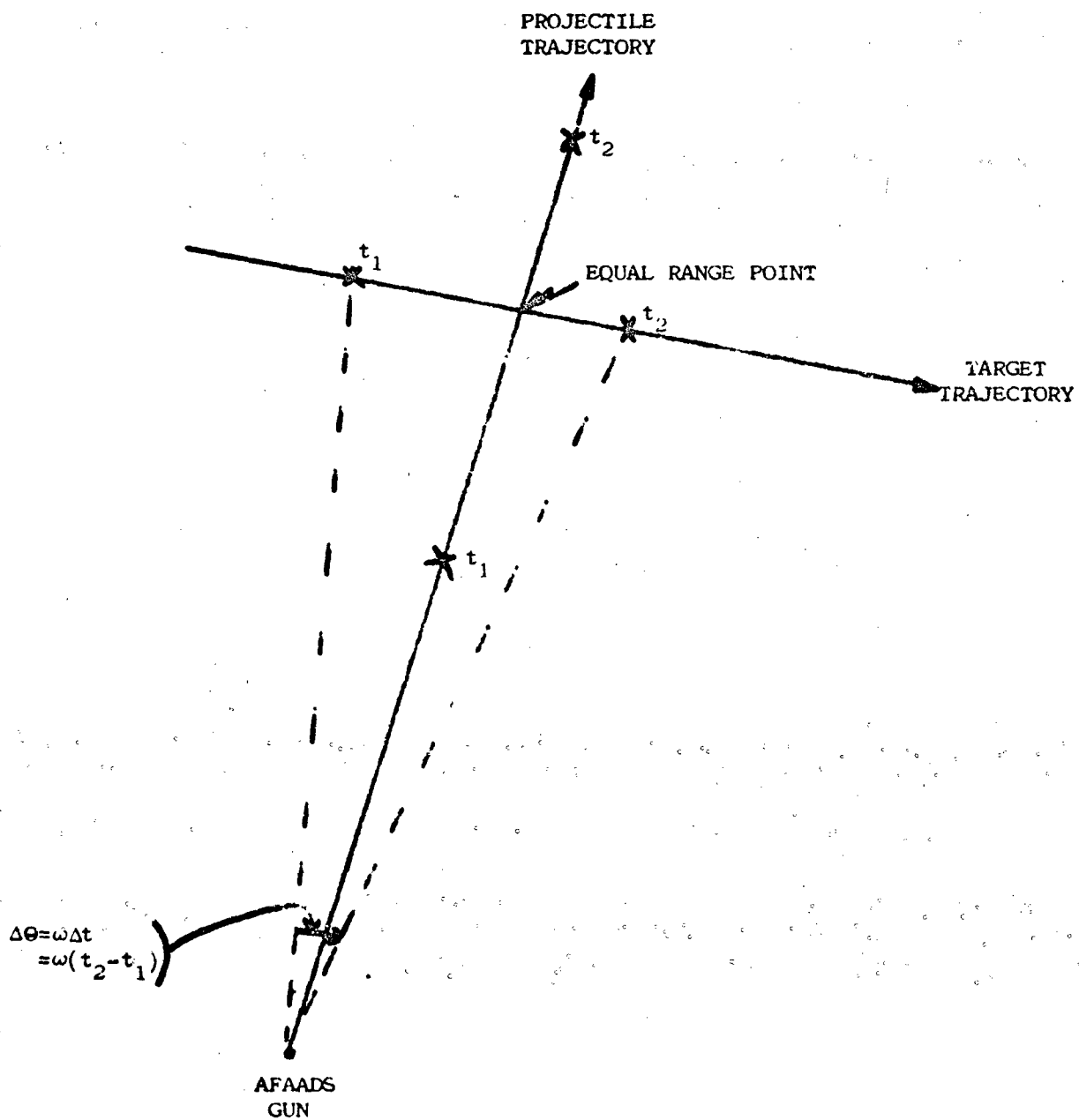


Figure D-1. Projectile Miss Distance Measurement Geometry

**D.3.2 Second Validity Criterion:** Figure D-1 assumes that the projectile is close to the target at the time that the miss distance measurements are made. However, for a directly approaching (or departing) target the projectile will always be in the field of view of the sensor from the moment of fire (neglecting any out of field-of-view effects due to the gravity correction factor). Thus, if the projectile-target distance is not within some limit, a bias correction could be made based upon data for the target and projectile being several kilometers apart. The gravity and wind corrections could then cause erroneous bias corrections. Also there would then be no way to sort out the several projectiles in flight.

It is thus proposed as a second validity criterion that the projectile-target range difference (range miss measurement) be less than the sum of (a) the range that can be traversed by the combination of projectile and target during one scan interval and (b) the range increment due to muzzle velocity dispersion

$$\Delta D(m) \leq (\dot{D}_p - \dot{D}_t) \Delta t + \Delta V_m t_p$$

where:  $\Delta D(m)$  = Magnitude of measured projectile range miss

$\dot{D}_p$  = Projectile velocity at target's range

$\dot{D}_t$  = Target opening velocity

$\Delta t$  = Scan period of laser

$\Delta V_m$  = Muzzle velocity dispersion

$t_p$  = Projectile flight time

Thus a range-gated FLIR system is required.

**D.3.3 Third Validity Criterion:** If range measurements are made on successive scans it is assumed that the same projectile is being tracked. Thus one measurement is made prior to the equal range point and the second one after this point. This third criterion is to select the smaller valid range measurement, and as a result, a factor of 2 can be used in the formula for the first criterion. (For Figure D-1, the second measurement would be selected.)

D.4 AFAADS Example: These criteria will now be applied to the AFAADS system firing a standard caliber round as used in the computer sizing task. For this task it is assumed that:

- a) FLIR/laser sampling period:  $\Delta t = 0.1$  second
- b) Angle miss distance measurement accuracy:  $\theta_0 = 0.25$  mil.
- c) Target closing velocity:  $\dot{D}_t = -300$  m/sec.
- d) Projectile velocity ( $\dot{D}_p$ ):
  - Muzzle 1175 m/sec
  - 4000 m range: 400 m/sec
- e) Muzzle velocity dispersion:  $\Delta V_m = 0.5\%$  muzzle velocity (6 m/sec)
- f) Projectile flight time:  $t_p = 6$  sec.

For these parameters,

- 1) Target angular velocity must be less than 5 mils/sec or 0.28 deg/sec.
- 2) Projectile range miss must be less than 106 m. at maximum range.

The implication of this first criterion on a specific tactical situation can be determined by considering a 300 m/sec target making a straight and level pass by the AFAADS gun.

The target is flying at 200 m and its point of closest approach is also 200 m. Against such a target, the AFAADS opens fire approximately 22 seconds before the point of closest approach. First projectile impact is at 4500 m and 15 seconds out. Projectile miss distance measurements can be made from this point in to a range of about 330 m (11 seconds out) when the angular rate exceeds the limits for the first criterion.

Similar situations exist for other tactical situations. In general valid or useful projectile miss measurements can be made at long range, in time to provide effective accurate fire when the target has closed.

## APPENDIX E

### SAMPLE BIAS CORRECTION ALGORITHM

A sample calculation is presented for the miss distance bias correction algorithm presented in Block XVI of Subsection 11.3. Reference should be made to this block for the definition of the several parameters evaluated.

#### E.1 Target Trajectory

A constant speed (300 m/sec) target is making a 45 deg. diving attack on a point close to the AFAADS gun. The target executes a 4-g pull-out maneuver such that the bottom of the pull-out occurs at the point-of-closest approach, namely at 200 m altitude and 200 m displacement from the gun (minimum slant range of 283 m).

During the time interval of 5.0 to 4.5 seconds before the point-of-closest approach, the target flight trajectories will have the values given in Table E-I.

#### E.2 AFAADS Gun Fire Parameters

During this same time period of 5.0 to 4.5 seconds before the point-of-closest approach, the AFAADS gun will be firing on the target and projectile miss distance measurements will be made. Table E-II lists the applicable gun parameters of:

- a) Projectile flight time
- b) Gun pointing angles
- c) Gun lead angles (difference between the target position (i.e., sensor pointing direction) and gun pointing angles)
- d) Projectile miss measurements.

Note that the projectile miss measurement values are an assumed random distribution, and do not necessarily represent the performance of an actual gun.

Table E-I Target Flight Parameters

Time		Target Parameters				
To Closest Point	Interval	Range		Azimuth		Elevation
t	j	D	$\dot{D}$	A	$\dot{A}$	E
(sec)	(numeric)	(m)	(m/sec)	(deg)	(deg/sec)	(deg)
5.0	1	1580	295	80.8	4	21.0
4.9	2	1550	295	80.7	4	20.8
4.8	3	1520	295	80.7	5	20.7
4.7	4	1490	295	80.6	5	20.5
4.6	5	1450	295	80.5	6	20.4
4.5	6	1430	295	80.4	7	20.3

Table E-II AFAADS Gun Fire Parameters

Time		Gun Parameters						Projectile	
To Closest Point	Interval	Projectile Flight Time	Gun Angles		Gun Lead Angles			Miss	
t	j	t(f)	A	E	Az( $\delta$ )	El( $\sigma$ )	Total(L)	$\Delta T$	$\Delta E$
(sec)	(numeric)	(sec)	(deg)	(deg)	(deg)	(deg)	(deg)	(deg)	(deg)
5.0	1	1.651	77.8	20.0	3.0	1.0	3.15	0.6	-0.5
4.9	2	1.615	77.6	20.0	3.1	0.8	3.19	1.2	0.0
4.8	3	1.569	77.3	20.0	3.4	0.7	3.47	-0.6	-0.3
4.7	4	1.531	77.0	20.0	3.6	0.5	3.63	0.9	-1.3
4.6	5	1.483	76.5	20.0	4.0	0.4	4.02	-0.7	0.4
4.5	6	1.459	76.0	20.0	4.4	0.3	4.4	1.0	0.3

## APPENDIX E (Continued)

### E.3 Bias Correction Constants

The several constants used in the bias correction algorithm, as defined in Items 4.1, 4.2, and 4.3 of Block XVI of Subsection 11.3, are listed in Table E-III.

TABLE E-III. BIAS CORRECTION CONSTANTS

<u>Parameter</u>			<u>Value</u>
Bias Variance:	Az	$\sigma$ (A)	10 mils
	El	$\sigma$ (E)	10 mils
	Vel	$\sigma$ (V)	8 mils
Projectile Vel:		V (a)	800 m/sec
Dispersion Variance:	Az	$\sigma$ (TW)	3 mils
	El	$\sigma$ (EW)	3 mils
	Vel	$\sigma$ (VW)	4 mils

### E.4 Defined Matrices

From Table E-I, Target Flight Parameters; Table E-II, Gun Fire Parameters, and Table E-III, Bias Correction Constants, three of the matrices defined in Block XVI of Subsection 11.3 can be evaluated. The results are presented in Table E-IV for the six data points between 5.0 and 4.5 seconds from the point-of-closest-approach.

### E.5 Sample Bias Corrections

The groundwork has now been laid for computing the actual bias correction values in azimuth, elevation, and muzzle velocity. Table E-V presents the results of going through three cycles of the algorithms. For the particular values used, the vector X(j) shows that the azimuth bias correction is slightly larger than the elevation correction. Both are about an order of magnitude greater than the muzzle velocity bias correction.

These calculations should be considered as illustrative, and not necessarily typical.

Table IV Defined Matrices

Time Interval	Matrices (Defined in Subsection 11.3 Block XVI)		
j	$Z(j) = \begin{pmatrix} \Delta T(j) \\ \Delta E(j) \end{pmatrix}$	$H(j) = \begin{pmatrix} \cos E & 0 & \cos E \sin \delta \\ 0 & 1 & \sin \sigma \end{pmatrix}$	$R(j)$
	(deg)	(numeric)	(mils) <sup>2</sup>
1	$\begin{pmatrix} 0.6 \\ -0.5 \end{pmatrix}$	$\begin{pmatrix} 0.934 & 0 & 0.0488 \\ 0 & 1 & 0.0174 \end{pmatrix}$	$\begin{pmatrix} 9.038 & 0.0136 \\ 0.0136 & 9.005 \end{pmatrix}$
2	$\begin{pmatrix} 1.2 \\ 0.0 \end{pmatrix}$	$\begin{pmatrix} 0.935 & 0 & 0.0505 \\ 0 & 1 & 0.0140 \end{pmatrix}$	$\begin{pmatrix} 9.041 & 0.0113 \\ 0.0113 & 9.003 \end{pmatrix}$
3	$\begin{pmatrix} -0.6 \\ -0.3 \end{pmatrix}$	$\begin{pmatrix} 0.935 & 0 & 0.0553 \\ 0 & 1 & 0.0122 \end{pmatrix}$	$\begin{pmatrix} 9.049 & 0.0108 \\ 0.0108 & 9.002 \end{pmatrix}$
4	$\begin{pmatrix} 0.9 \\ -1.3 \end{pmatrix}$	$\begin{pmatrix} 0.937 & 0 & 0.0588 \\ 0 & 1 & 0.0087 \end{pmatrix}$	$\begin{pmatrix} 9.055 & 8.16 \times 10^{-3} \\ 0.00816 & 9.001 \end{pmatrix}$
5	$\begin{pmatrix} -0.7 \\ 0.4 \end{pmatrix}$	$\begin{pmatrix} 0.937 & 0 & 0.0655 \\ 0 & 1 & 0.0070 \end{pmatrix}$	$\begin{pmatrix} 9.069 & 7.34 \times 10^{-3} \\ 7.34 \times 10^{-3} & 9.001 \end{pmatrix}$
6	$\begin{pmatrix} 1.0 \\ 0.3 \end{pmatrix}$	$\begin{pmatrix} 0.938 & 0 & 0.0719 \\ 0 & 1 & 0.0050 \end{pmatrix}$	$\begin{pmatrix} 9.083 & 5.75 \times 10^{-3} \\ 5.75 \times 10^{-3} & 9.000 \end{pmatrix}$

Table E-V Sample Bias Corrections

Matrices (Defined in Subsection 11.3 Block XVI)					
Time Interval	$K(j)$ (Numeric)	$M(j)^2$ (Mils) <sup>2</sup>	$U(j)$ (Deg.)	$X(j)$ (Deg.)	
$j$					
1	$\begin{pmatrix} 0.969 & -6.03 \times 10^{-4} \\ -6.46 \times 10^{-4} & 0.918 \\ 3.24 \times 10^{-2} & 1.018 \times 10^{-2} \end{pmatrix}$	$\begin{pmatrix} 100 & 0 & 0 \\ 0 & 100 & 0 \\ 0 & 0 & 100 \end{pmatrix}$	$\begin{pmatrix} -0.580 \\ 0.459 \\ -0.0143 \end{pmatrix}$	$\begin{pmatrix} -0.580 \\ +0.459 \\ -0.0143 \end{pmatrix}$	
2	$\begin{pmatrix} 0.515 & 9.7 \times 10^{-4} \\ 3.12 \times 10^{-3} & 0.476 \\ 2.31 \times 10^{-2} & -7.30 \times 10^{-3} \end{pmatrix}$	$\begin{pmatrix} 9.7 & 6.0 \times 10^{-2} & -3.02 \\ 0.181 & 8.2 & -1.014 \\ -3.02 & -1.020 & 63.9 \end{pmatrix}$	$\begin{pmatrix} -0.339 \\ -0.220 \\ -1.83 \times 10^{-3} \end{pmatrix}$	$\begin{pmatrix} -0.919 \\ +0.239 \\ -0.0161 \end{pmatrix}$	
3	$\begin{pmatrix} 0.342 & -1.078 \times 10^{-3} \\ 1.067 \times 10^{-2} & 0.323 \\ 3.80 \times 10^{-2} & 1.80 \times 10^{-2} \end{pmatrix}$	$\begin{pmatrix} 5.1 & 0.017 & -33.23 \\ 0.117 & 4.3 & -9.58 \times 10^{-3} \\ -3.23 & -1.020 & 63.9 \end{pmatrix}$	$\begin{pmatrix} 0.500 \\ 3.52 \times 10^{-2} \\ 5.66 \times 10^{-2} \end{pmatrix}$	$\begin{pmatrix} -0.419 \\ +0.204 \\ +0.0405 \end{pmatrix}$	

## APPENDIX F

### BALLISTIC CORRECTION FOR GRAVITY DROP

#### F.1 Summary

This appendix examines simple algorithms for computing the super-elevation angle required in the ballistic computation to compensate for gravity drop.

It appears that one may compute superelevation angle  $\phi_s$  as

$$\phi(s) = [g/2V(0)] [t(p) + c_1 t^2(p) + \dots] \cos E(p) \quad (F-1)$$

where:  $E(p)$  = Predicted target elevation angle at the aim point

$V(0)$  = Muzzle velocity

$t(p)$  = Predicted time of flight

$c(j)$  = Constants for a given weapon under "standard conditions" but may be a function of muzzle velocity under non-standard conditions. (For the Oerlikon gun  $c(1)$  equals 0.0465 per second).

The predicted time of flight,  $t(p)$ , is computed separately from the ballistic equations (for instance, a fifth order polynomial is used in Section 11.3 Block IV) as a function of the predicted slant range,  $D(p)$ .

The reason for computing  $\phi(s)$  as a series in  $t(p)$  rather than  $D(p)$  is the empirical observation that the simple cosine dependence on  $E(p)$  applies only for constant  $t(p)$ , not for constant  $D(p)$ . This observation is based upon the Oerlikon gun ballistic tables.

#### F.2 Introduction

Superelevation is the added elevation angle applied to a gun to account for gravity drop of the projectile. For vacuum and flat earth trajectories, gravity drop is just  $(1/2) g t^2(p)$  where  $t(p)$  is the projectile's time of flight. Now let  $\phi$  be the gun quadrant elevation angle including the gravity correction and  $\phi(s)$  be the superelevation angle. Then by the law of sines (see Figure F-1),

$$\sin \phi(s) = (1/2) g t^2(p)/D(p) \cos \phi \quad (F-2)$$

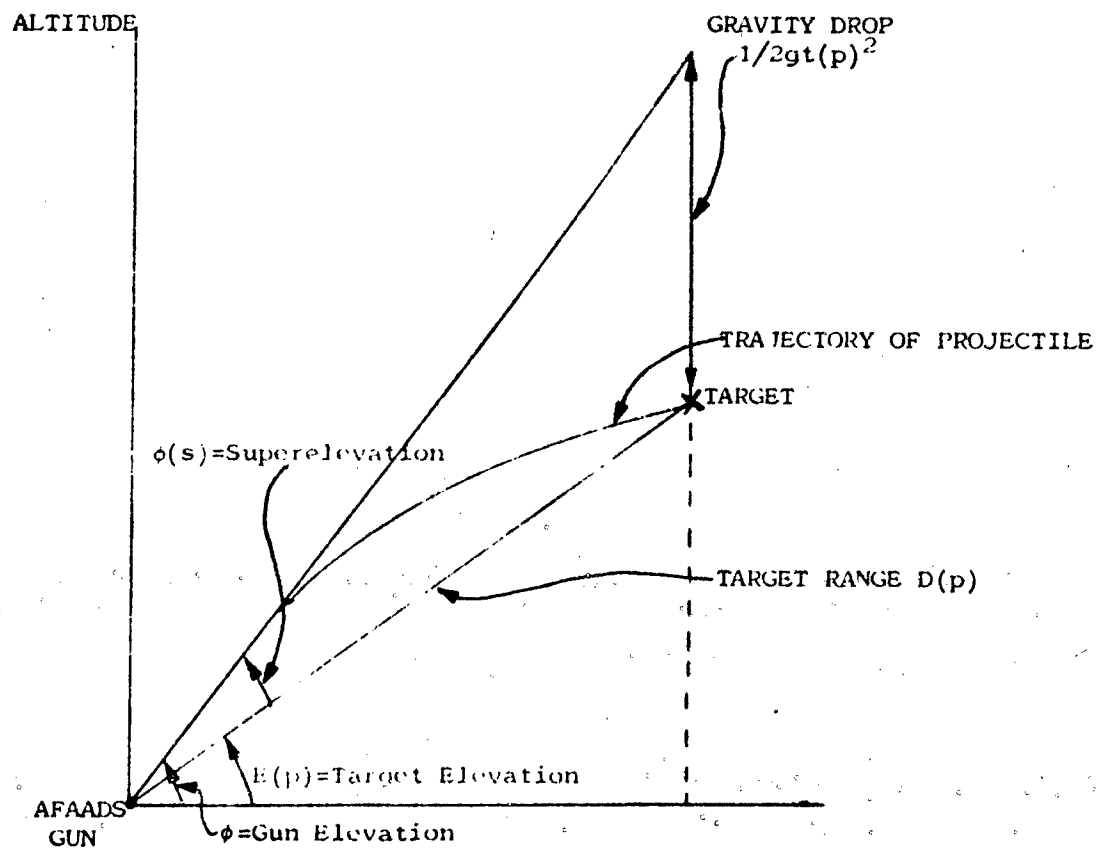


Figure F-1. Geometry for Gravity Drop Correction

For antiaircraft firing,  $\phi(s)$  will always be small. Then,  $\phi$  will be approximately equal to  $E(p)$ , the sight or elevation angle to the aim point. We can then save one iterative operation by approximating  $\phi(s)$  in terms of  $E(p)$  rather than  $\phi$ . Since

$$\phi = E(p) + \phi(s) \quad (F-3)$$

then for small values

$$\phi = E(p) \quad (F-4)$$

The basic computation in the ballistic unit is the computation of the time of flight  $t(p)$  for specified  $D(p)$  and  $E(p)$ . It may be possible to use some of the terms of the  $t(p)$  computation in the  $\phi(s)$  computation, and this should be kept in mind when the time comes to reduce the set of ballistic algorithms down to a minimum number of operations.

However, at this point we assume that we have as inputs to the  $\phi(s)$  computation:  $E(p)$ ,  $D(p)$ , and  $t(p)$ , the last from the ballistic algorithm. Using the Oerlikon ballistics, Figure F-2 shows  $\phi(s)$  plotted both against  $D(p)$  and against  $t(p)$  for  $E(p) = 0$  deg. The  $\phi(s)$  versus  $t(p)$  curve has less curvature, and we shall show later that  $t(p)$  is a preferred parameter for another reason.

### F.3 Suggest Algorithms

For short times of flight,

$$D(p) = V(o) t(p) \quad (F-5)$$

where  $V(o)$  is the muzzle velocity. Substituting this equation and equation F-4 into equation (F-2), we have

$$\phi(s) = [g/2V(o)] t(p) \cos E(p) \quad (F-6)$$

This is the asymptotic value of the superelevation angle for short times of flight. Using this as the leading term in a polynomial approximation, the superelevation angle can be expressed as,

$$\phi(s) = [g/2V(o)] t(p) \cos E(p) [1 + c(1) t(p) + c(2) t^2(p) + \dots] \quad (F-7)$$

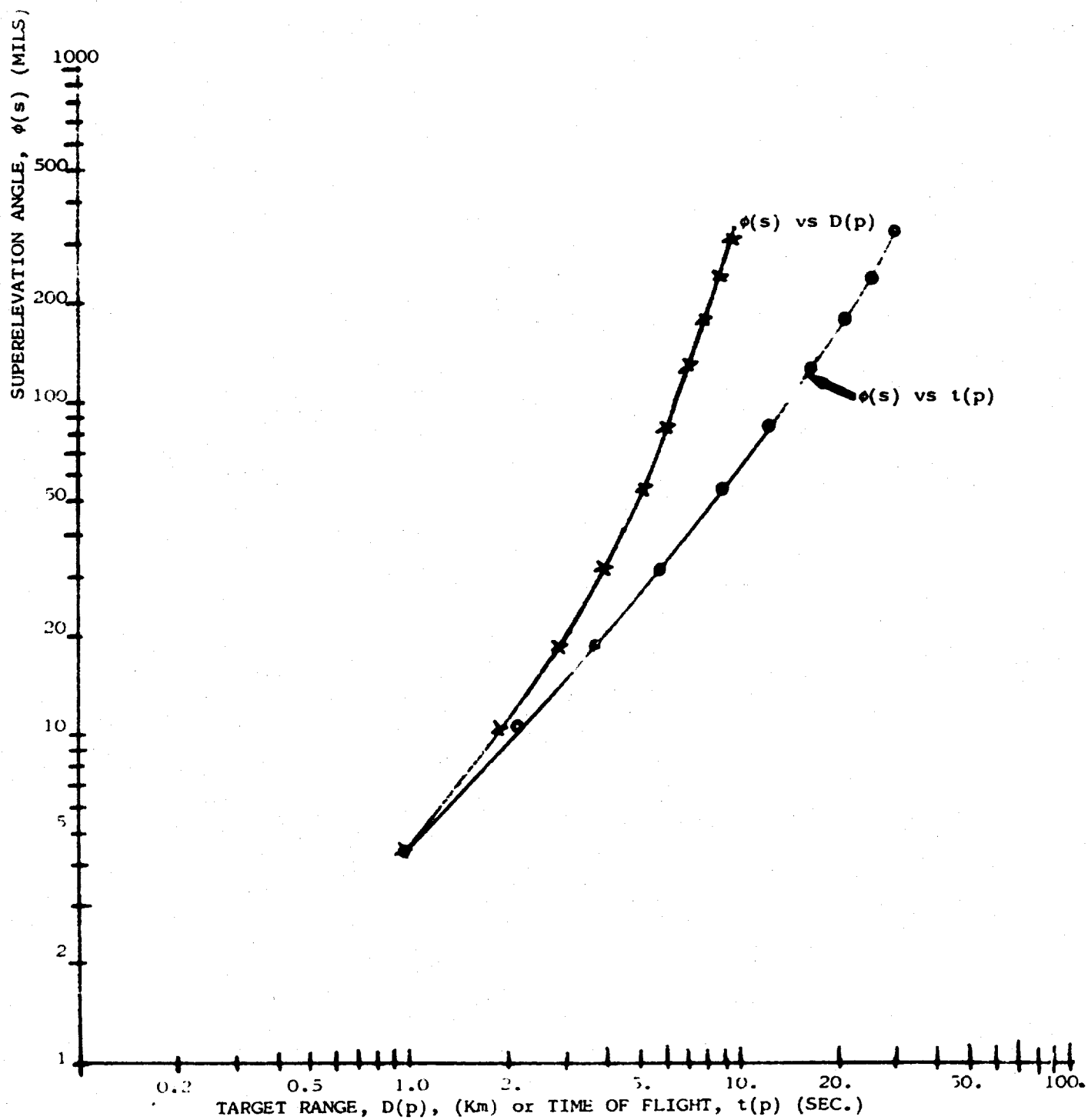


Figure F-2. Oerlikon Superelevation Angle Versus Range and Flight Time

The objective is to obtain an expression for  $\phi(s)$  accurate to 0.3 mil out to a range of 3km relative to the Oerlikon ballistic firing table data.

Thus we expect the superelevation angle to vary as  $\cos E(p)$  at a given range  $D(p)$ . To test this the Oerlikon data is plotted in Figure F-3 at 3km range in the form of  $\phi(s)$  versus  $\cos E(p)$ . From the figure, we see the relationship is approximately a straight line, as expected; but the line does not go through the origin, contrary to the predicted relationship given in equation F-7. To see how close to a straight line this relationship actually is, we plot in Figure F-4  $\phi(s)/\cos E(p)$  versus  $E(p)$ . Although there is considerable scatter, there is also a trend of the form

$$\phi(s) = \phi(s,0) \cos E(p) [1 + a \cos E(p)] \quad (F-8)$$

where:  $\phi(s,0)$  is the superelevation angle for  $\cos E(p)$  equal to zero (i. e., a target directly overhead)  
 $a$  is a constant.

However, on referring to the Oerlikon ballistic tables, we note that at 3km range, the time of flight is less for high trajectories than for low trajectories. Also  $\phi(s)/\cos E(p)$  is lower for high trajectories than for low trajectories to the same point.

This suggests that we may get a better simple relation if we compute

$$\phi(s) = \cos E(p) f[t(p)] \quad (F-9)$$

From the firing tables we interpolate  $\phi(s)$  to a constant time of flight of 3.780 seconds by the formula.

$$\phi(s,3.780) = \phi(s,t) - [t(p) - 3.780] [\Delta\phi(s)/\Delta t(p)] \quad (F-10)$$

This time of 3.780 seconds corresponds to the time of flight at 3km range and zero elevation angle. Figure F-5 shows the comparative results of plotting  $\phi(s)/\cos E(p)$  for constant time of flight  $t(p)$  of 3.780 seconds and for constant range of 3km. (The latter is a repeat of the data in Figure F-4). From the figure, we can see that curve for constant time of flight is essentially a constant, to the accuracy of the firing table point-to-point scatter. In fact if we choose an average value of 18.8 for  $\phi(s)/\cos E(p)$  and a 3.780 time of flight, we obtain  $\phi(s)$  for all  $E(p)$  to an accuracy of better than 0.3 mil.

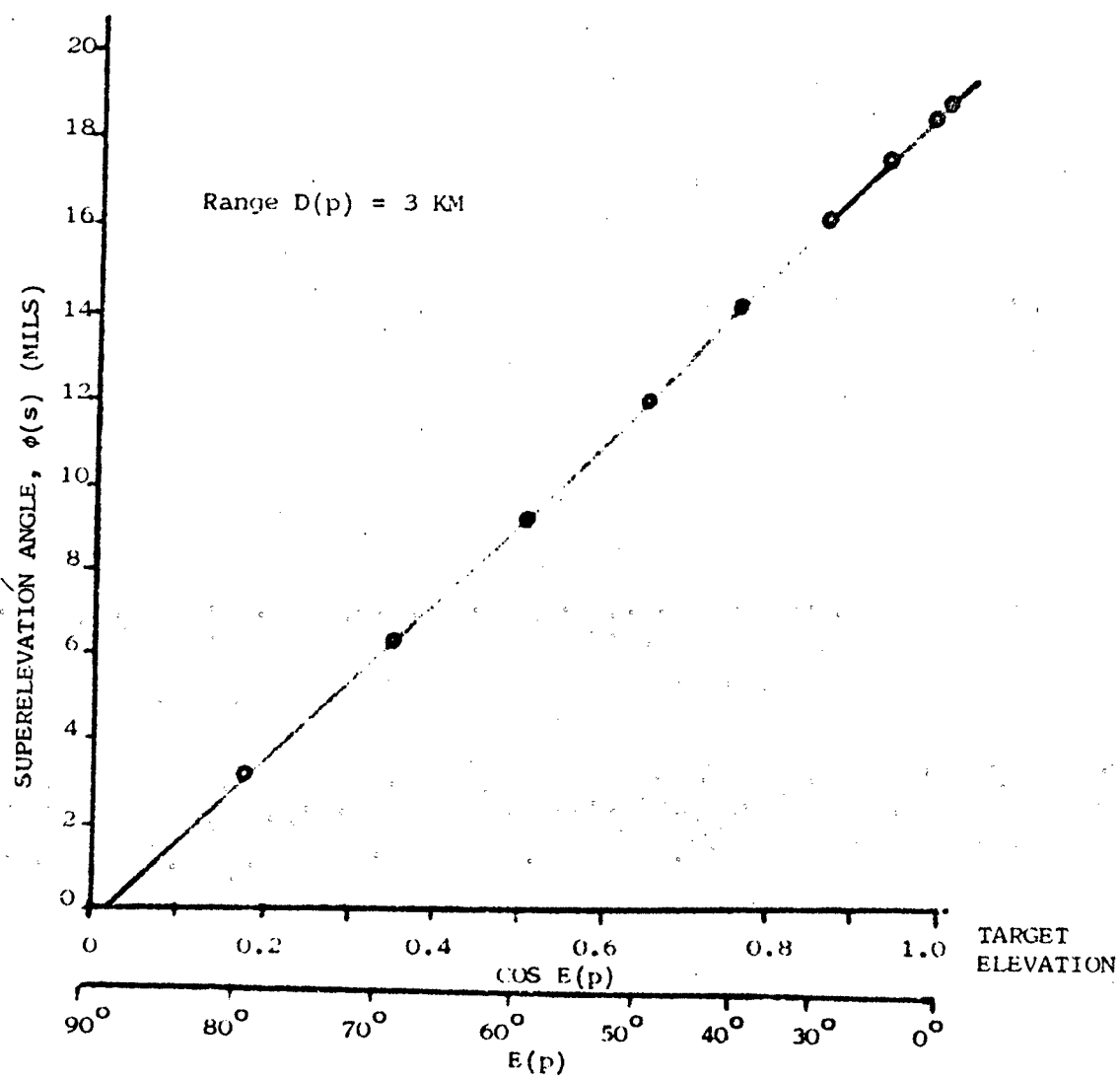


Figure F-3. Superelevation Angle Vs. Target Elevation Angle

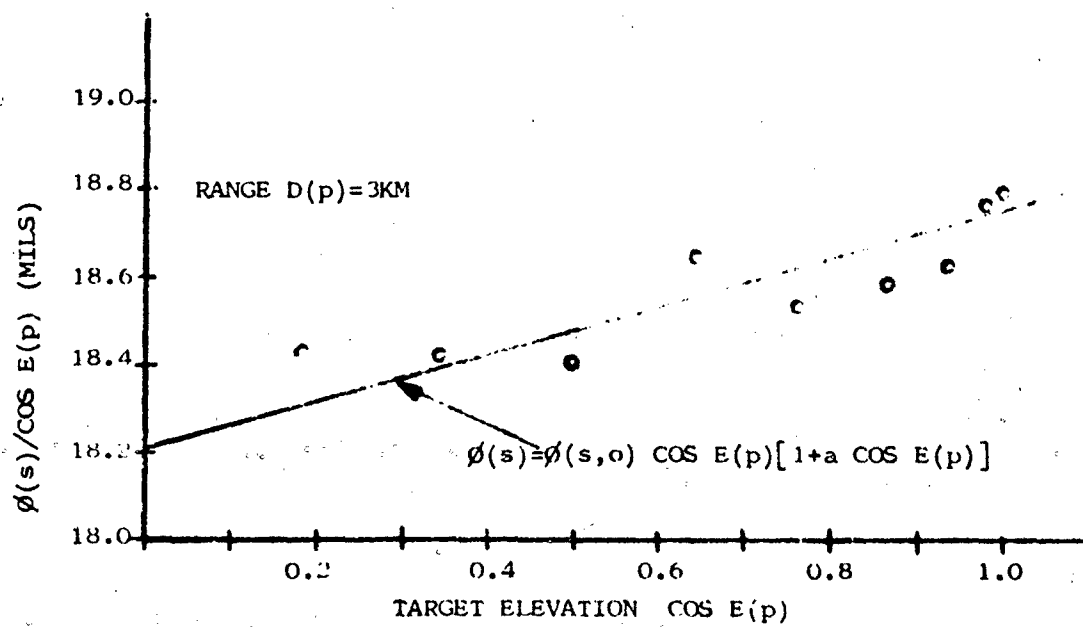


Figure F-4.  $\phi(s)/\cos E(p)$  Vs.  $\cos E(p)$  at Constant Range

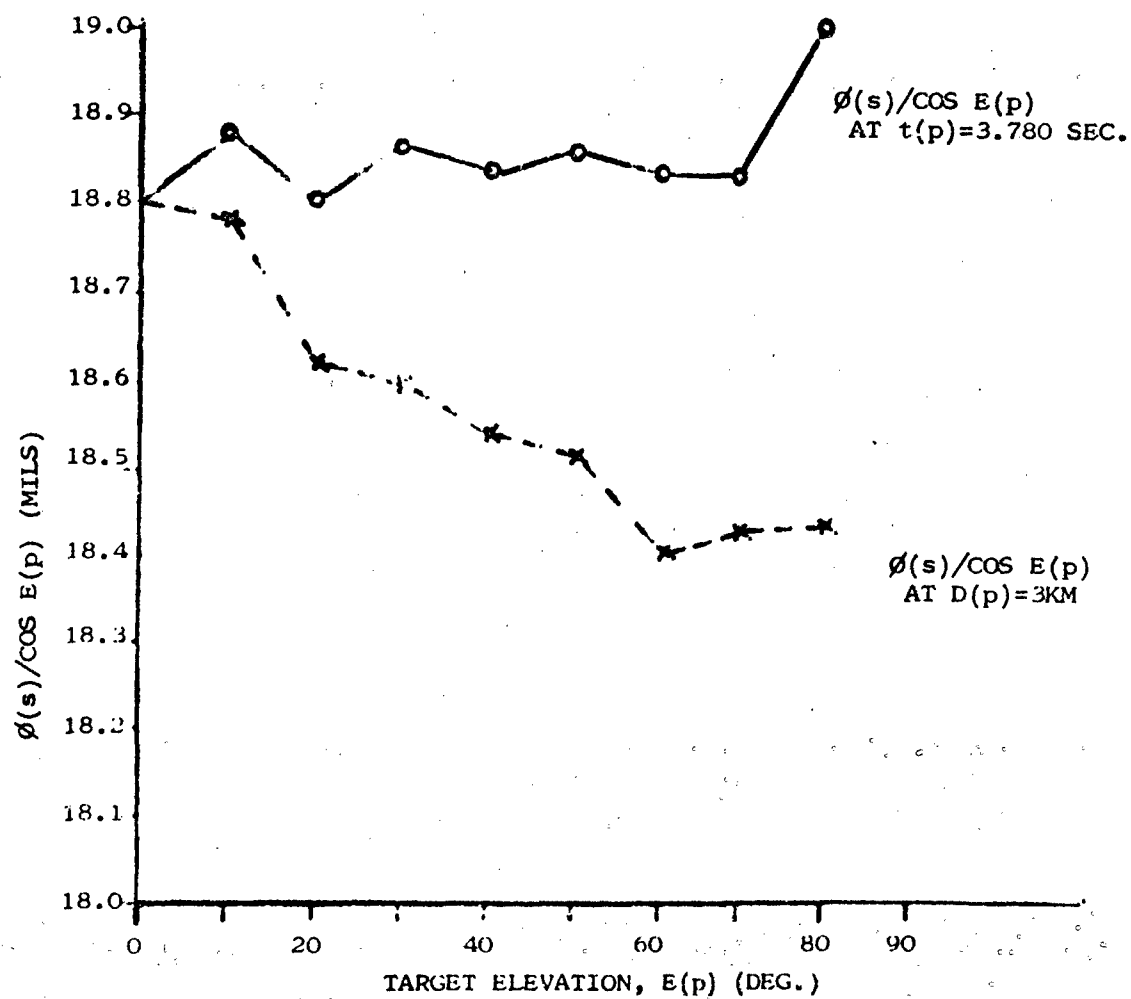


Figure F-5. Superelevation Relationships  
at Constant Flight Time and Constant Range

#### F.4 Selected Algorithm

Thus we see that  $\phi(s)$  should be expanded in terms of  $t(p)$  rather than  $D(p)$ ; as was indicated by equations F-6 and F-7, for a given target elevation angle  $E(p)$ . Figure F-6 is a plot of  $\phi(s)/t(p)$  versus  $t(p)$  at zero target elevation angle. This graph indicates:

$$\phi(s)/t(p) = [g/2V(o)] [1 + c(1) t(p) + \dots] \quad (F-11)$$

should provide a satisfactory solution out to about 6 seconds of flight. This is the equation given in the Introduction to this Appendix (equation F-1).

From the graph the ordinate intercept gives 4.22 mils/sec for  $\phi(s)/t(p)$ , i. e., the value of  $g/2V(o)$ . The slope gives a value of  $c(1)$  equal to 0.0465 per second.

This simple approach seems better than an alternate of the form

$$\phi(s) = \frac{t(p)g/[2V(o)]}{[1 - d(1)t(p) + d(2) t^2(p) + \dots]} \quad (F-12)$$

for if we write the preceding expression as

$$Y = [2V(o)/g - t(p)/\phi(s)]/t(p) = b(1) - b(2)t(p) + \dots \quad (F-13)$$

where  $b(j)$  is  $2V(o)/g$  times  $c(j)$

we obtain  $Y$  vs  $t(p)$  as shown in Figure F-7.

However, both forms should be investigated in finer detail before making a choice.

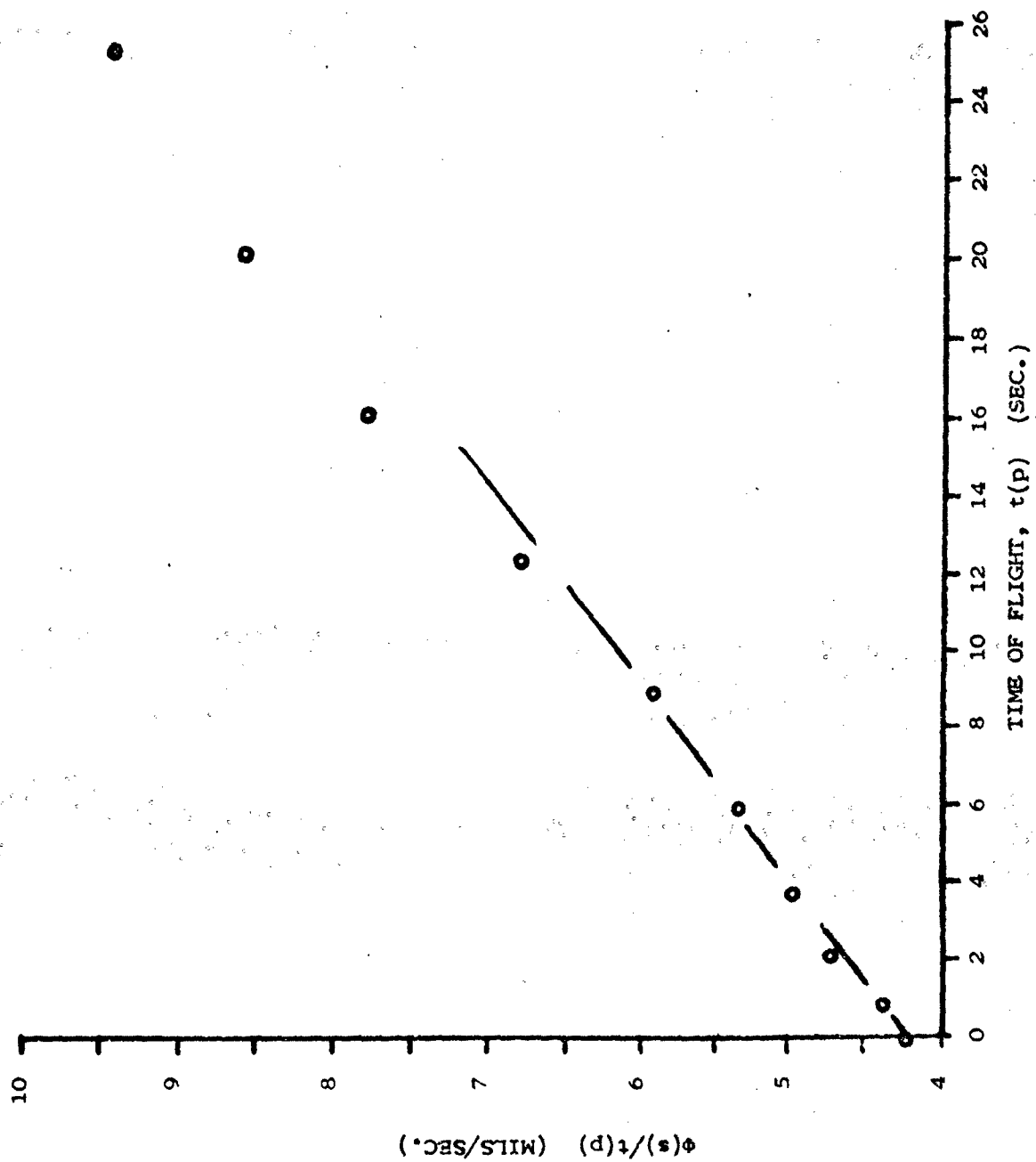


Figure F-6. Superelevation Vs. Time of Flight

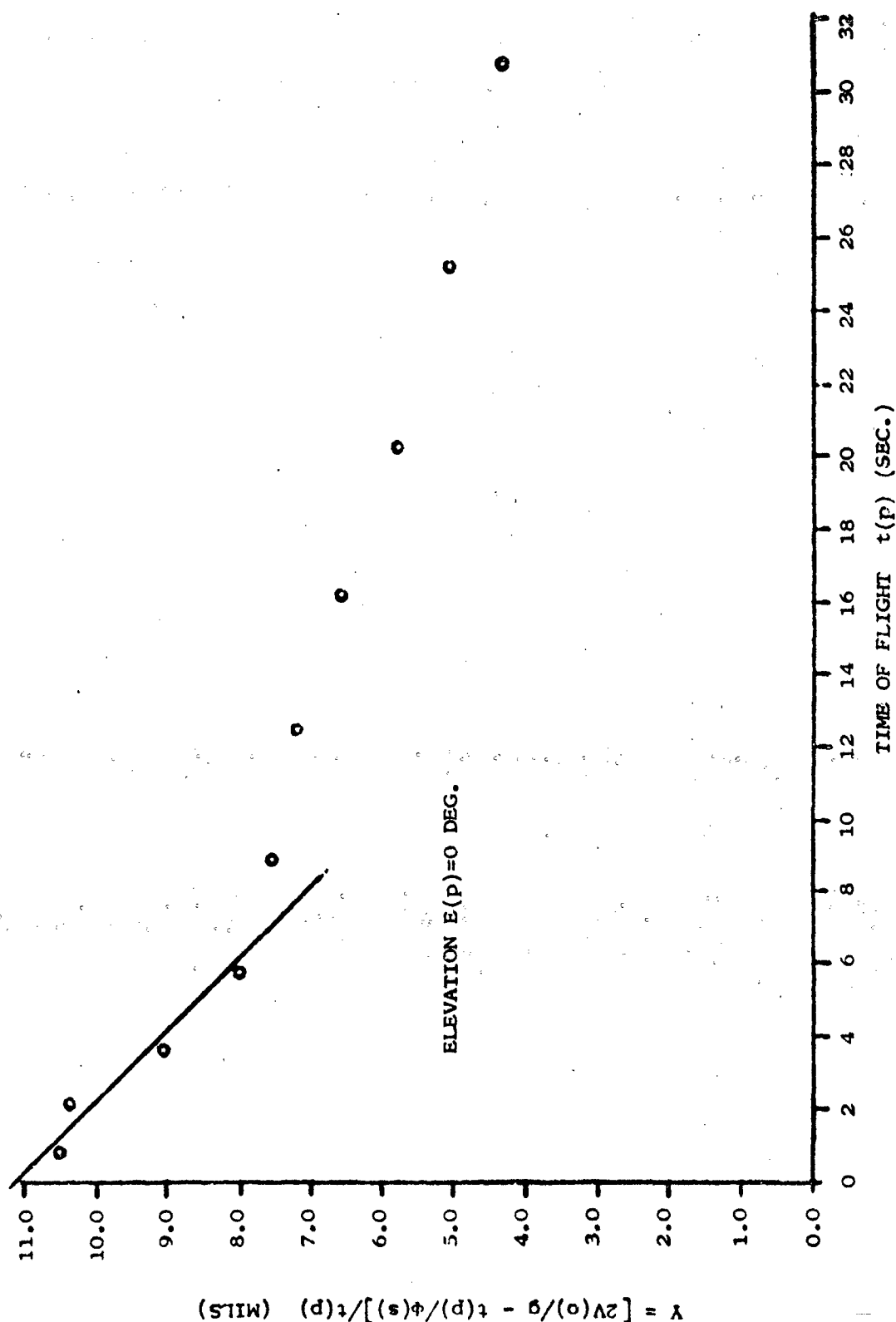


Figure F-7. Alternate Form for Superelevation Correction

## APPENDIX G

### BALLISTIC CORRECTION COMPUTATION FOR CROSS WIND

The "drift" correction for a 10 m/s cross wind from the Oerlikon firing tables is shown in Figure G-1.

To develop a computational algorithm, the first obvious step is to correct for the secant of the elevation angle  $E(p)$ , by multiplying by  $\cos E(p)$ . It is shown in Part B of Figure G-2 that  $\Delta L \cos E(p)$  varies with  $E(p)$  where  $\Delta L$  is the drift. The reason is the effect of changing air density on the trajectory.

Density varies about as

$$\rho = \rho_o e^{-kh} \quad (G-1)$$

where  $H$  is the altitude. In terms of slant range, we have

$$\rho = \rho_o e^{-Dk \sin E(p)} \quad (G-2)$$

Taking this factor into account, Part A of Figure G-2 shows that:

$\Delta L \cos E(p)$  plots (at 3km) as a linear function of  $\sin E(p)$ .

At the maximum range of 6km for which data is given the fit is not as good (Figure G-3), and the slope is steeper. The slope, in fact varies very roughly with time of flight, rather than slant range.

Returning to the variation of  $\Delta L$  itself with range, we see (Figure G-4) that it is well represented as proportional to time of flight out to 6 seconds time of flight (or 4km). Hence we propose the following algorithm

$$\Delta L = K W(c) t(p) \sec E(p) [1 - k t(p) \sin E(p)]$$

where:  $\Delta L$  is the drift correction in mils

$W(c)$  is the cross wind

$t(p)$  is the time of flight

$E(p)$  is the target elevation angle

From Figure G-4 the two constants can be evaluated:

$$K = 0.1153 \text{ mils/meter}$$

$$k = 0.0293/\text{sec}$$

These are the values used in the fire control algorithms of Section 11.3.

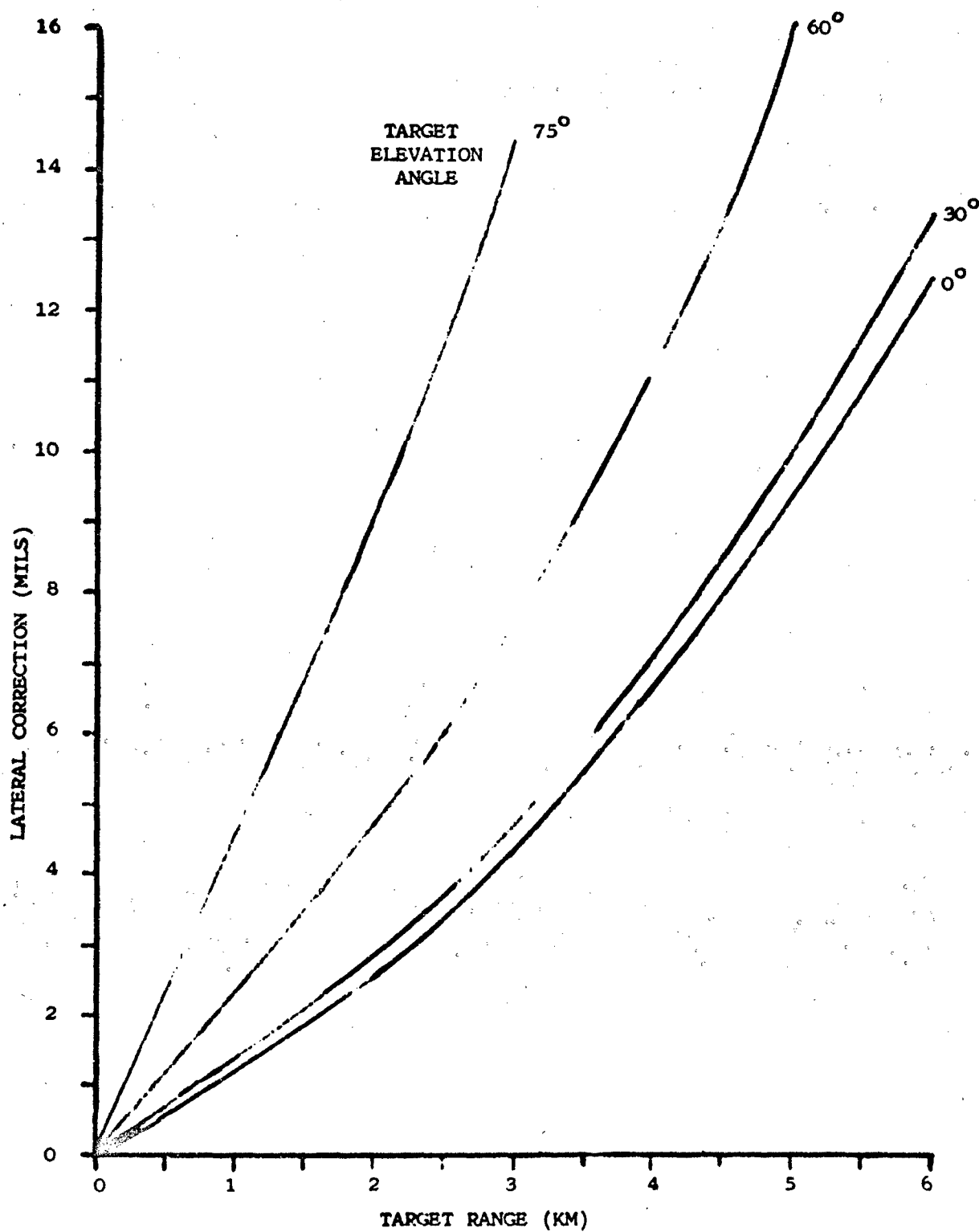


Figure G-1. Drift Correction for 10 m/sec. Cross Wind

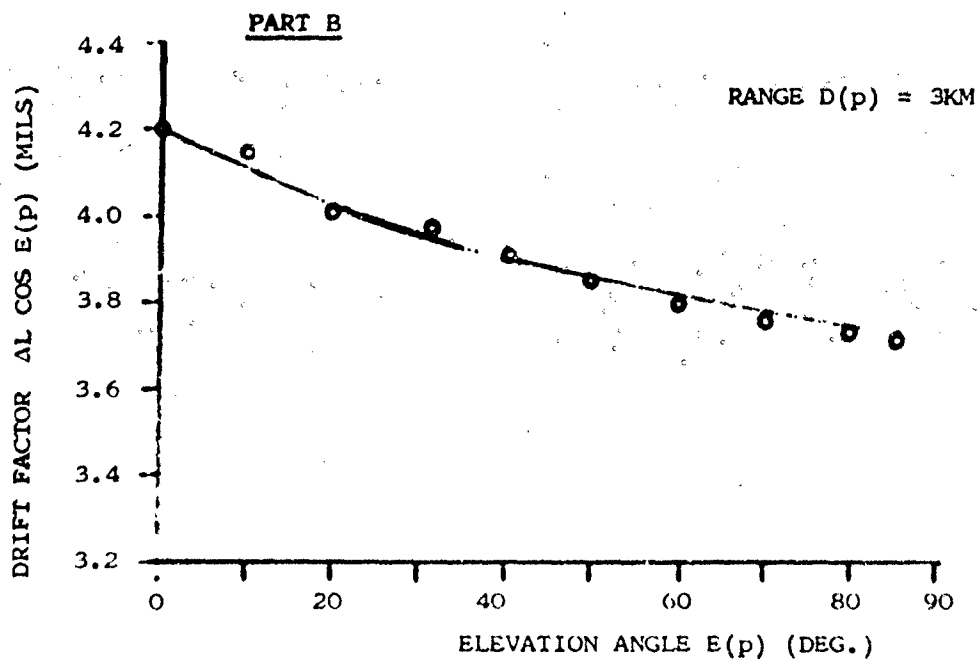
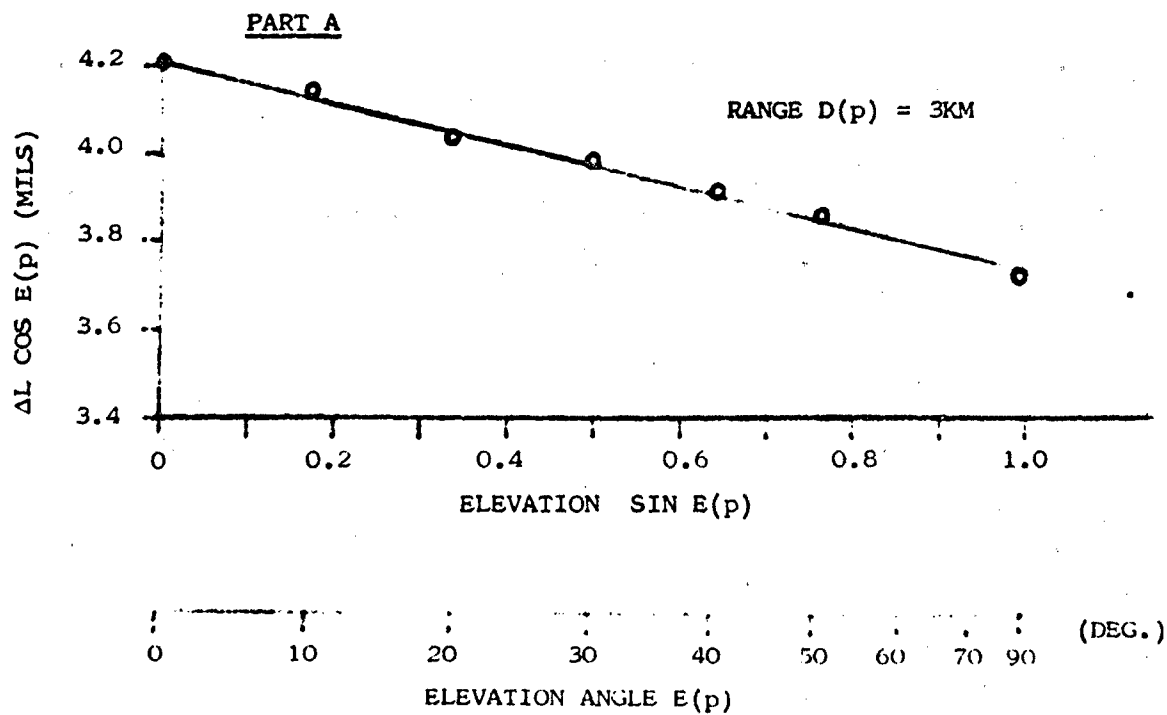


Figure G-2. Drift Correction Factor Vs. Elevation Angle

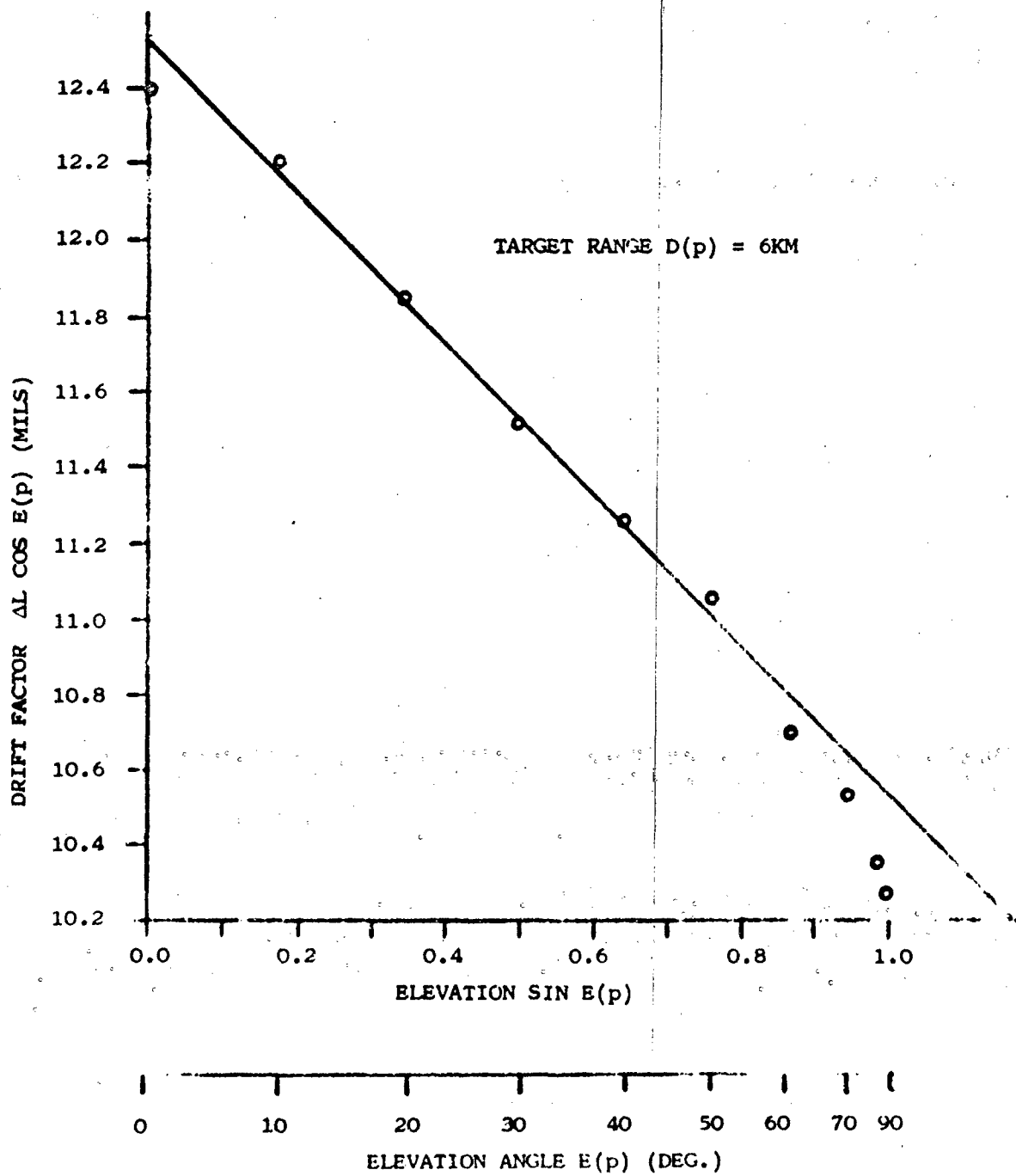


Figure G-3. Maximum Range Draft Correction Factor

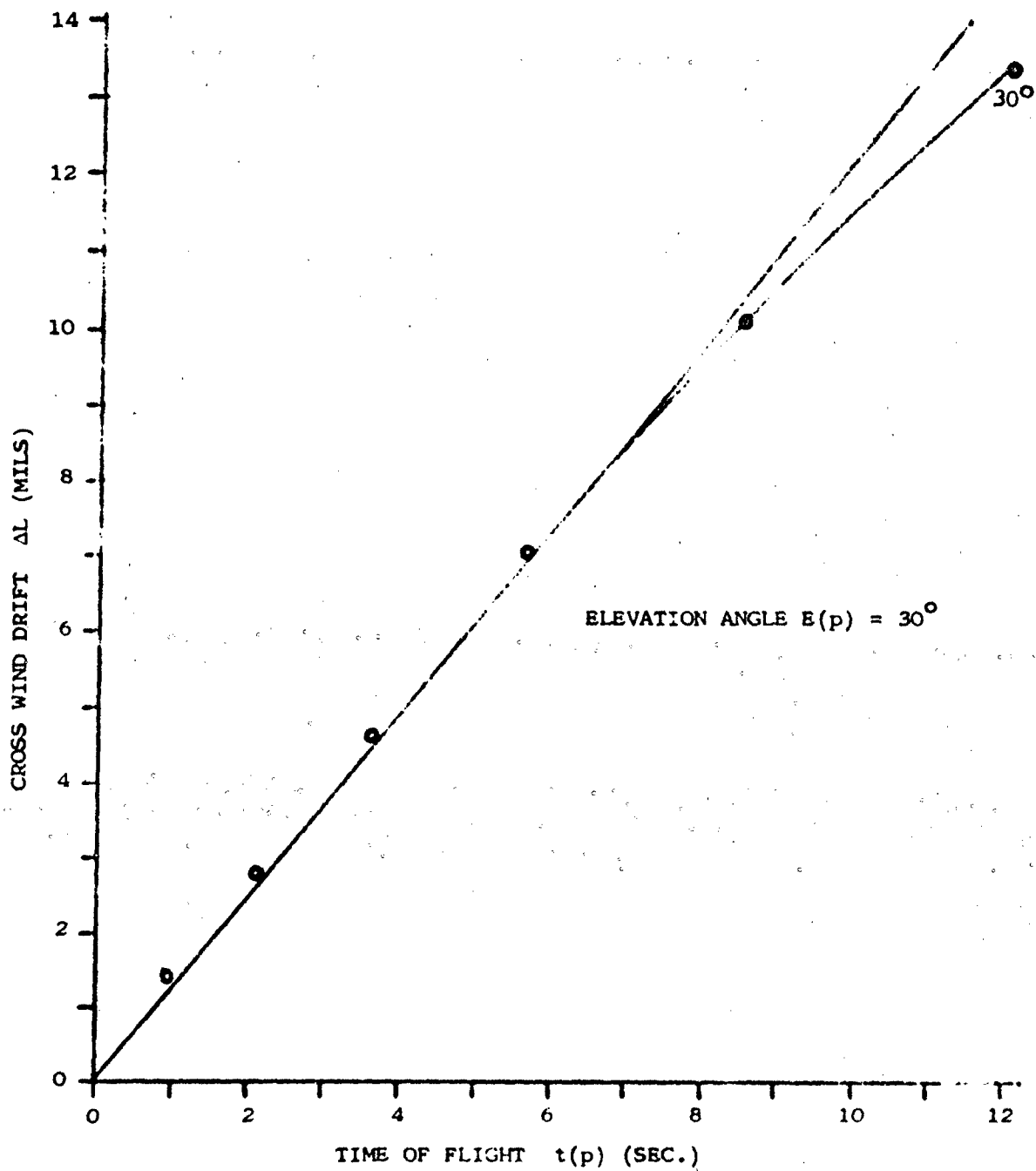


Figure G-4. Drift Correction Vs. Time of Flight

Considering the uncertainty in cross wind  $W_c$ , this should be adequate.  
Before building a system, of course, more detailed analysis should be done.

Best Available Copy

Best Available Copy

## APPENDIX H

### TABLE OF ACRONYMS

AA	Antiaircraft
ADA	Air Defense Artillery
AFAADS	Advanced Forward Area Air Defense System
ALU	Arithmetic Logic Unit
EMI	Electromagnetic Interference
FCP	Fire Control Program
FLIR	Forward Looking Infrared
FOV	Field of View
I/O	Input/Output
IOCP	Input-Output Control Program
IOU	Input-Output Unit
LED	Light Emitting Diode
LOS	Line of Sight
LSB	Least Significant Bit
MSI	Medium Scale Integration
PROM	Programmable Read Only Memory
PSC	Programming Support Center
RAM	Random Access Memory
ROM	Read Only Memory
TO & E	Table of Organization and Equipment

Best Available Copy

## APPENDIX I

### REFERENCES

1. Final Report, a Parametric Study of the Advanced Forward Area Air Defense Weapon System (AFAADS) Vol. I, System Analysis. Data Systems Division, Litton Systems, Inc. Report MS 00678, 2 October 1970.
2. Final Report, Supplement to Parametric Study of Advanced Forward Area Air Defense Weapon System (AFAADS), Vol. I, Analysis. Data Systems Division, Litton Systems, Inc., Report TD 20871, 15 September 1972. (AD-903-969L)
3. AD 651-988 (Unclassified)
4. Norman Morrison - Introduction to Sequential Smoothing and Prediction, Chap. 7, "The Fixed-Memory Polynomial Fitter," McGraw Hill 1969.
5. The Oerlikon - Contraves 35mm AA Tank - Some Operational Considerations, International Defense Review, Vol. 5, No. 4, pp 377-383 (August 1972).

Best Available Copy

END

DATE

FILMED

6-74



